ARBEITSBERICHTE
DES INSTITUTS FÜR MATHEMATISCHE MASCHINEN UND DATENVERARBEITUNG (INFORMATIK)
FRIEDRICH ALEXANDER UNIVERSITÄT ERLANGEN NÜRNBERG
HERAUSGEBER: H. Billing, A. Bode, W. Händler, U. Herzog, F. Hofmann, K. Leeb, P. Mertens, H. Müller, G. Nees, H. Niemann, H. J. Schneider, D. Seitzer, H. Wedekind

DREIDIMENSIONALE MODELLBILDUNG IN DER BILDANALYSE

Peter Heß

DISSERTATION
IER 1 · ERLANGEN · JANUAR 1987

Disbui. Erlangen
1986

UNIV.
LEIDEN
BIBL.

ISSN 0344 - 3515

Digitized by the Internet Archive
in 2026

https://archive.org/details/IA41538118_0012

DREIDIMENSIONALE MODELLBILDUNG IN DER BILDANALYSE

Der Technischen Fakultät

der Universität Erlangen-Nürnberg

zur Erlangung des Grades

DOKTOR - INGENIEUR

vorgelegt

von

Peter Heß

ERLANGEN 1987

Als Dissertation genehmigt von der
Technischen Fakultät der
Universität Erlangen - Nürnberg

Tag der Einreichung:	14. Juli 1986
Tag der Promotion:	4. Dezember 1986
Dekan:	Prof. Dr. H. Brand
Berichterstatter:	Prof. Dr. H. Niemann
	Prof. Dr. H. Wedekind

Für die Förderung und Betreuung dieser Dissertation gilt Herrn Prof. Dr. Niemann mein besonderer Dank, ebenso wie Herrn Prof. Dr. Wedekind für die Übernahme des Zweitgutachtens.

Daneben erinnere ich mich gerne an die Zusammenarbeit mit Kollegen und Studenten während der Zeit, in der diese Arbeit entstand. Ihnen allen sei hier gedankt.

I N H A L T

1. Einleitung 1

2. Grundlagen 5
2. 1 Ein Bildanalysesystem 5
2. 2 Benötigtes Wissen 9
2. 2. 1 Problemunabhängiges Wissen 9
2. 2. 2 Problemabhängiges Wissen 10
2. 2. 3 Metawissen 11
2. 3 Theoretische Grundlegung des Modellbegriffs 12
2. 4 Geometrische 3D-Transformationen 14

3. Ansätze zur Modellbildung für visuelle Daten 18
3. 1 Grundsätzliche Überlegungen zur Modellbildung für visuelle Daten 18
3. 2 Kriterien für die Modellbildung 20
3. 2. 1 Formale Kriterien für die geometrische Modellbildung 20
3. 2. 2 Informelle Kriterien für die geometrische Modellbildung 21
3. 2. 3 Kriterien für die Bildanalyse 22
3. 3 Ansätze zur Modellbildung 24
3. 3. 1 Übersicht 24
3. 3. 2 Ansätze zur 2D-Modellbildung 26
3. 3. 3 Ansätze zur 3D-Modellbildung 29
3. 4 Anwendungsbeispiele 36
3. 4. 1 CAD - Systeme 37
3. 4. 2 Computeranimation 39
3. 4. 3 Bildanalysesysteme 39

4. Ansätze zur Wissensrepräsentation 44
4. 1 Grundlagen und Ziele der Wissensrepräsentation 44
4. 2 Verfahren der Wissensrepräsentation 45
4. 3 Einsatz und Vergleich verschiedener Ansätze zur Wissensrepräsentation 49

5. Ein Bildanalysemodellierungssystem 51
5. 1 Anforderungen an ein Bildanalysemodellierungssystem 52
5. 2 3D-Modellierung mit verallgemeinerten Zylindern 53
5. 2. 1 Definition der Primelemente 53
5. 2. 2 Definition von Objekten 55
5. 2. 3 Definition einer Szene 58
5. 3 Oberflächenattribute 60
5. 4 Bildung von Objektklassen 61
5. 4. 1 Arten der Klassenbildung 62
5. 4. 2 Fuzzy-Funktionen 64
5. 4. 3 Objektklassenbildung mit Fuzzy-Funktionen 67
5. 5 Bedingungen 69
5. 6 Syntax der Modellbildung 71
5. 7 Darstellung auf dem Rechner 77
5. 7. 1 Verwendete relationale Datenbank 77
5. 7. 2 Darstellung des Modells durch Relationen 81

6. Ein anderer Ansatz für ein Bildanalysemodellierungssystem 84
6. 1 Modellbildung 84
6. 2 Darstellung auf dem Rechner 86
6. 3 Vergleich der beiden Ansätze 87

7. Zur Analyse von Bildern 91
7. 1 Bewertungsfunktion 93
7. 2 Kontrollalgorithmus 95

8. Zwei Beispiele 100
8. 1 Modellerstellung 100
8. 2 Erzeugung von Testbildern 102
8. 3 Natürliche Szenen 103
8. 3. 1 Allgemeine Vorstadtszene 103
8. 3. 2 Ein Blick aus dem Fenster 111
8. 4 Industrielle Szenen 113

9. Zusammenfassung 119

Literatur 124

Anhang

1. Einleitung

"Sehende" Computer verarbeiten Bilder, die mit einer Kamera aufgenommen werden, um daraus eine geeignete Beschreibung zur Lösung eines gegebenen Problems zu erzeugen. Solche Bildanalysesysteme werden in vielen Bereichen eingesetzt. Darunter fallen u.a. (vgl. /NIE 80/) Schriftzeichenerkennung, Medizin, Industrie, Erdfernerkundung, Hochenergiephysik und Militär. Was heute mit Bildanalysesystemen erreicht wird, soll an drei Beispielen aus den Bereichen Medizin, Industrie und Militär erläutert werden.

Medizin

Am Lehrstuhl für Informatik 5 (Mustererkennung) wird in Zusammenarbeit mit dem Institut für Nuklearmedizin der Universität Erlangen-Nürnberg ein System zur Interpretation von Bildsequenzen des menschlichen Herzens entwickelt /HOF 86, BUN 83/. Ziel des Systems ist die Diagnose von Bewegungsabnormalitäten des Herzens, die auf eine krankhafte Veränderung hindeuten. Dazu werden die erkannten Befunde zur Unterstützung des Arztes in einen diagnostischen Zusammenhang gestellt. Folgende Vorgehensweise wurde gewählt: Zur Aufnahme wird dem Patienten radioaktives Jod injiziert. Dann erzeugt man mit einer Gammakamera eine in 12 Bilder aufgeteilte Bildfolge, die genau einen Herzzyklus umfaßt. Aus dieser Bildfolge detektiert man die Kontur der linken Herzkammer und teilt den Bereich der Herzkammer in vier bzw. acht Sektoren ein. Die Bild-zu-Bild-Änderungen der Größe der Herzkammer werden zu einem Bewegungsprofil zusammengefaßt. Da man den normalen Bewegungsverlauf eines gesunden Herzens gespeichert hat, ist es möglich, diesen mit dem aus der Bildfolge gewonnenen Bewegungsprofil zu vergleichen. Aus diesem Vergleich lassen sich dann Bewegungsabnormalitäten ableiten, die mit Wissen über Krankheitssymptome zu möglichen Diagnosen erweitert werden können. Sagerer /SAG 85/ gibt folgende Ergebnisse für 21 getestete Bildsequenzen: Drei der 21 Bildsequenzen wurden vom System als nicht klassifizierbar zurückgewiesen. Bei drei weiteren Bildsequenzen gibt es leichte Abweichungen zwischen Arzt und System bei der Beurteilung. Diese Abweichungen sind aber sehr gering. Die restlichen 15 Bildsequenzen wurden von Arzt und System übereinstimmend diagnostiziert.

Dieses Bildanalysesystem zeigt, daß heutige Systeme in einem begrenzten Anwendungsbereich bereits sehr gute Ergebnisse liefern können, auch wenn bis zu einem klinischen Einsatz noch weitere Tests und Verbesserungen notwendig sind.

Industrie

Für den industriellen Einsatz kann man heute schon einige Bildverarbeitungssysteme kaufen. Eine Übersicht darüber findet sich z.B. bei Winkler /WIK 85/. Diese Systeme werden vor allem zur Qualitätsprüfung oder zum Erkennen von Teilen eingesetzt. Aus der Menge der Systeme zur Qualitätsprüfung erläutern wir ein von Perkins /PER 84/ vorgestelltes System, das allerdings (unseres Wissens nach) nicht gekauft werden kann. Es prüft bei bestückten Leiterplatten die Vollständigkeit und in begrenztem Umfang auch die Korrektheit der Bestückung. Dazu besteht es aus zwei Programmen, einem Programm MODELFORMER, mit dem die Modelle der Leiterplatten eingelernt werden und einem Programm INSPECTOR, mit dem die Korrektheit einer Leiterplatte anhand des vorher eingegebenen Modells überprüft wird. Das Modell einer Leiterplatte unterscheidet zwei Ebenen. Auf der Leiterplattenebene wird die Leiterplatte anhand von Kontrollpunkten örtlich justiert, und mit weiteren Kontrollpunkten wird das Vorhandensein der sogenannten Komponenten, z.B. Widerstände, Chips usw. inspiziert. Auf der Komponentenebene können dann für einzelne Komponenten noch weitere Tests definiert werden, die aufgrund von verschiedenen Grauwertstatistiken eine Unterscheidung in gute und schlechte Komponenten ermöglichen. Das gesamte Modell der zu untersuchenden Leiterplatte muß von einem Bediener interaktiv eingegeben werden, bevor dann das Programm INSPECTOR automatisch die definierte Testfolge ausführt. Im Ganzen sind Systeme, die sich im industriellen Einsatz befinden, meist noch auf die entsprechende Anwendung zugeschnitten und dabei abhängig von konstanten Umgebungsbedingungen. Allerdings sind Bildverarbeitungssysteme gerade bei der Qualitätskontrolle schon ziemlich erfolgreich /SCF 85/.

Militär

Weite Verbreitung finden bildverarbeitende Systeme auch in der modernen Waffentechnologie. Als Beispiel sei ein schon ziemlich altes Verfahren erläutert, das in den Marschflugkörpern (cruise missiles) eingebaut ist /TSI 83, REI 85/. Dieses System heißt TERCOM (terrain-contour-matching) und erlaubt eine Zielgenauigkeit von 100 m. Das Problem bei derartigen Raketen besteht darin, daß durch Wettereinflüsse oder durch unregelmäßiges Arbeiten der Triebwerke Abweichungen vom einprogrammierten Kurs auftreten können. Um diese auszugleichen, sind im TERCOM-System Höhenkarten für rechteckige Gebiete von z.B. 2 km x 10 km gespeichert, in denen für jeden Geländeausschnitt von 100 m x 100 m die Durchschnittshöhe angegeben ist. Überfliegt nun die Rakete ein solches Gebiet, so mißt ein eingebauter Radar die Höhe des Geländes und kann dann durch einen Vergleich mit der gespeicherten Karte eine Kurskorrektur vornehmen. Wie unmittelbar zu erkennen ist, funktioniert TERCOM nur über einer hügeligen Land-

schaft, weshalb inzwischen auch schon Nachfolgesysteme entwickelt wurden, die andere Geländemerkmale verwenden und dadurch auch für andere Geländeformen geeignet sind. Was man aber bereits an TERCOM sieht, ist die Tatsache, daß die prinzipiell gleichen Algorithmen und Programme in vielen sehr unterschiedlichen Anwendungen zu Lösungen führen können und dann auch eingesetzt werden.

Aus diesen drei Anwendungsbeispielen ist bereits ersichtlich, daß es im Bereich der Bildverarbeitung schon einige Systeme gibt. Diese sind aber meist darauf angelegt, einzelne spezielle Probleme zu lösen. Ist der Problembereich komplexer, so existieren bisher nur Ansätze und noch keine Gesamtlösungen. Das Gleiche trifft auf allgemeine Bildananlysesystem zu, die ein breites Spektrum von Problemen bearbeiten sollen. Ein solches allgemeines Bildanalysesystem ist noch nicht in Sicht.

Ziel in der Bildananlyse ist die automatische Auswertung optischer Daten, wobei das Ergebnis der Analyse eine für die Aufgabenstellung geeignete Beschreibung /NIE 81/ der im Bild vorhandenen Information sein sollte. Dabei ist das mit einer Kamera aufgenommene Bild Ausgangspunkt der Verarbeitung, die dann mit einem Computer erfolgt. Diese Verarbeitung darf man aber nicht mit dem Sehprozeß beim Menschen gleichsetzen, da im Computer eine reine Auswertung der optischen Daten aufgrund der implementierten Programme erfolgt. Langenheder /LAN 86/ schreibt dazu:

"Computer haben weder Motivationen noch Emotionen. Sie besitzen nichts, das auch nur annähernd an das herankommt, was menschliches Bewußtsein ausmacht. Computer können also wirklich nicht sehen oder hören, wenn man die mit diesen Begriffen verbundenen Bedeutungsinhalte ernst nimmt. Das aber sollte man tun, insbesondere wenn man ansonsten als Vertreter der "harten Wissenschaften" den Anspruch erhebt, eine präzise Sprache mit allgemein verbindlichen, eindeutigen Begriffen zu verwenden."

Deswegen werden wir nur von Bildanalyse und Bildverarbeitung sowie von Robotern mit einem bilderkennenden System sprechen.

Diese Arbeit gliedert sich in zwei Teile. Die Kapitel 2, 3 und 4 geben eine Übersicht über verschiedene Ansätze zur dreidimensionalen Modellbildung. Dabei beginnen wir mit der Darlegung von Grundlagen zur Bildanalyse und zur Modellbildung in Kapitel 2. Kapitel 3 folgt mit einer ausführlichen Beschreibung von Kriterien und Ansätzen, die im computerunterstützten Konstruieren (CAD) und in der Bildanalyse zur Generierung eines geeigneten Modells verwendet werden. Den Abschluß des ersten Teils bildet eine kurze Darstellung wichtiger Formalismen zur Repräsentation der Modelle auf dem Rechner. In den Kapiteln 5, 6, 7 und 8

wird dann ein eigener Ansatz zur dreidimensionalen Modellbildung für die Bildanalyse vorgestellt. Aufbauend auf sog. verallgemeinerten Zylindern wird der Ansatz in Kapitel 5 eingehend besprochen und in Kapitel 6 einem vom CAD geprägten Ansatz gegenübergestellt. Kapitel 7 erläutert Möglichkeiten zur Analyse von Bildern anhand dieses Modells und gibt einen Kontrollalgorithmus an, der eine solche Analyse durchführen kann. Kapitel 8 beschäftigt sich mit der Modellerstellung und diskutiert Beispiele des gesamten Prozesses von der Modellerstellung bis zur Analyse. Dabei betrachten wir die Problemkreise "natürliche Szenen" und "industrielle Szenen". Eine kurze Zusammenfassung beschließt die Arbeit.

2. Grundlagen

Nachdem in der Einleitung beispielhaft einige Anwendungen der Bildanalyse vorgestellt wurden, werden in diesem Kapitel die Grundlagen für die Erörterung der Modellbildung in der Bildanalyse gelegt. Ausgehend von den Zielen der Bildanalyse werden die Vorgehensweise und die notwendigen Systemkomponenten in der Bildanalyse diskutiert. Um einschätzen zu können, welches Wissen zur Modellbildung in der Bildanalyse benötigt wird, schließt sich eine Klassifizierung des Wissens im Hinblick auf Bildanalyseprobleme an. Eine theoretische Grundlegung des Begriffs Modellbildung folgt, bevor das Kapitel 2 mit einer kurzen Einführung in geometrische 3D-Transformationen abschließt.

2.1 Ein Bildanalysesystem

Ziel der Bildanlyse ist die Gewinnung einer für die Lösung eines Problems angemessenen Beschreibung, die, abhängig von der gestellten Aufgabe, sehr verschieden aussehen kann. Drei mögliche Beschreibungen eines Bildes von einem Werkstück (siehe Bild 2.1) lauten z.B. folgendermaßen:

1. Auf dem Bild sind 20 Flächen gleichen Grauwerts zu erkennen, dabei umschließt die Fläche 1 alle anderen Flächen.
2. Auf dem Bild ist ein Rotor eines Elektromotors zu erkennen. Der Schwerpunkt des Rotors liegt im Punkt (x=255, y=250).
3. Auf dem Bild ist ein Rotor eines Elektromotors zu erkennen. Für die Montage kann er so nicht gegriffen werden. Er muß zuerst um 180^{o} um die lokale x-Achse gedreht werden. Dazu muß der Rotor um 10cm angehoben werden. Die Greifpunkte liegen an den Stellen ...

Für einen Einsatz in der Fabrik wird die erste Beschreibung meist ungeeignet sein, während die zweite und die dritte Beschreibung je nach Aufgabenstellung eine geeignete Problemlösung darstellen können.

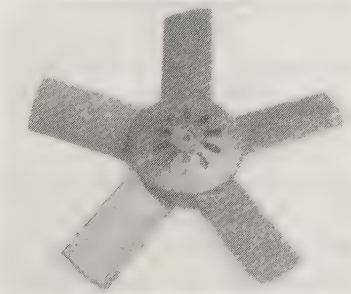


Bild 2.1: Ein Werkstück

Um eine geeignete Beschreibung aus einem Bild zu erhalten, empfiehlt sich ein stufenweises Vorgehen, wie es bei Niemann /NIE 85a/ ausführlich diskutiert wird. Ähnlich geht auch Marr /MAR 82b/ mit den Stufen "primal sketch", "2 1/2 D-sketch" und 3D-Interpretation vor. Bild 2.2 zeigt den schematischen Aufbau eines Bildanalysesystems, anhand dessen wir die einzelnen Verarbeitungsschritte erklären wollen. Das System besteht aus den sechs Blöcken Prozesse, Ergebnisse, Modell, Kontrolle, Erklärung und Lernen, die jeweils noch in einzelne Module gegliedert sein können.

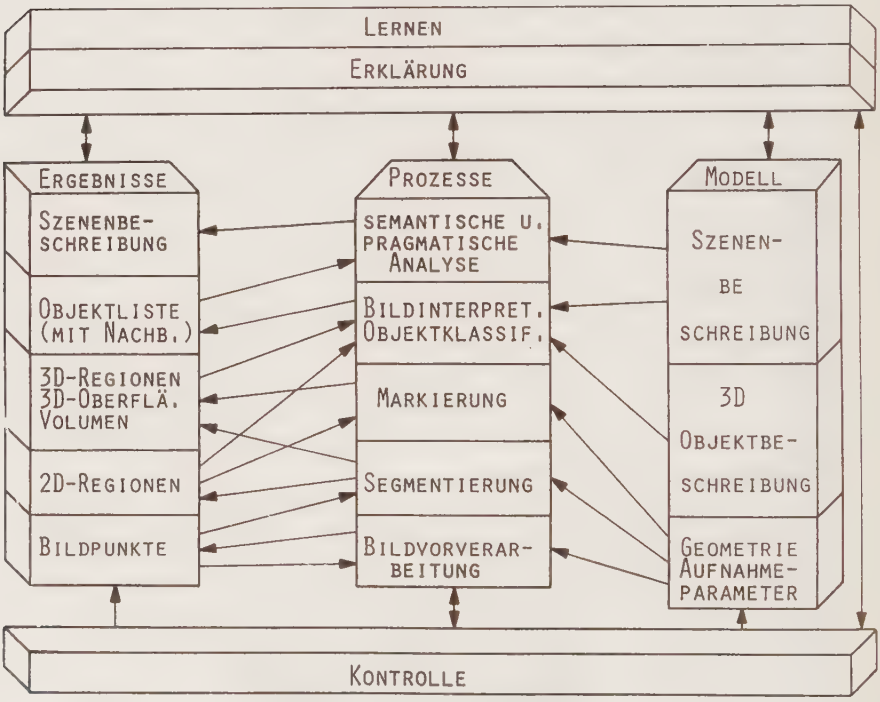


Bild 2.2: Blockbild eines Bildanalysesystems

Zur Aufnahme benutzt man einen optisch-elektrischen Wandler, meist eine Fernsehkamera, die zunehmend auf der Basis eines CCD-Chips aufgebaut ist (siehe z.B. /FAI 84/). Das Signal, das man aus der Kamera erhält, wird abgetastet,

analog-digital umgesetzt (siehe z.B. /SEI 77/) und in einen Bildspeicher des Rechners gebracht, auf dem die weitere Verarbeitung abläuft. Damit liegt das Bild als ein Feld von standardmäßig 512x512 Bildpunkten im Rechner vor. Jeder Bildpunkt hat dabei einen Wert zwischen 0 und 255, falls das Bild ein Grauwertbild ist oder drei Werte jeweils aus diesem Bereich bei einem Farbbild. Bildfolgen werden hier außer Acht gelassen (vergleiche hierzu z.B. /NAL 85a/). Eine genauere Betrachtung der Erzeugung eines Bildes findet sich bei Ballard und Brown /BAL 82/ oder Levine /LEV 85/. Das Feld mit den Bildpunkten ist die unterste Schicht der Ergebnisse.

Erstes Teilziel ist das Finden von 2D-Regionen. Dies sind Gebiete im Bild, die zusammenhängen und gleiches oder ähnliches Aussehen haben bezüglich irgendeines Homogenitätskriteriums. Solche Homogenitätskriterien sind Grauwert, Farbe oder Textur. Auch das duale Vorgehen, also die Suche nach Sprüngen bezüglich dieser Kriterien, kann zum Erreichen dieses Teilziels führen. Alle Segmentierungsalgorithmen (vgl. z.B. /HAA 85/) verwenden immer die Annahme, daß homogene Regionen jeweils zu einem Objekt gehören, während sich an den Objektgrenzen ein Sprung im Homogenitätskriterium ergibt. Da bei realen Bildern die Verhältnisse meist nicht so einfach liegen, kann man durch geeignete digitale Filterung oder andere Algorithmen (wie z.B. Histogrammglättung) versuchen, das Bild für die weitere Verarbeitung zu verbessern. Diese Verfahren, die ein Feld von Bildpunkten als Ergebnis liefern, sind im Block Bildvorverarbeitung zusammengefaßt (vgl. z.B. /GON 77/). Das Ergebnis bei der Bearbeitung dieses Teilziels kann eine Beschreibung wie Nummer 1 oben sein, die allerdings kaum ausreichen wird zur Lösung einer Aufgabenstellung.

Das nächste Teilziel wird deshalb die Erzeugung einer Liste der im Bild vorhandenen Objekte und ihrer örtlichen Beziehungen sein. Es müssen also Fragen der Art:

- Zu welchen Objekten passen die im Bild gefundenen Regionen?
- Welche Regionen gehören zu einem Objekt?
- Welche Objekte befinden sich im Bild?
- Welche Lage haben diese Objekte im Bild?

geklärt werden.
Dazu benötigt man ein Modell der Objekte, die im Bild zu erwarten sind. Diese werden nun mit den im Bild gefundenen Strukturen (z.B. Regionen) verglichen und die beste gefundene Übereinstimmung ergibt dann die Interpretation des Bildes. Für diesen Vergleich benötigt man eine Bewertungsfunktion und eine Vorgehensstrategie, da aus Zeitgründen nicht alle möglichen Vergleiche Bild-Modell

durchgeführt werden können. Diese Vorgehensstrategie orientiert sich oft an Suchverfahren aus der künstlichen Intelligenz (vgl. z.B. /NIL 82/). Kapitel 7 dieser Arbeit geht nochmals auf dieses Problem ein. Ein Zwischenschritt zu diesem Teilziel kann eine Verbesserung der im Bild gefundenen Strukturen sein. Es gibt heute Algorithmen, die versuchen, die bei der Aufnahme verloren gegangene dritte Dimension wieder zu rekonstruieren. Diese Verfahren laufen unter dem Schlagwort "Shape from"-Algorithmen (vgl. z.B. /BAL 82/). Erzeugt man zu den Bildpunkten jeweils die Tiefe oder die Oberflächenkrümmung an diesem Bildpunkt, so läßt sich ein Homogenitätskriterium der Art kontinuierliche Tiefe oder kontinuierliche Oberflächenkrümmung verwenden, um eine Segmentierung des Bildes in 3D-Regionen und 3D-Oberflächen zu gewinnen. Diese Strukturen können dann wieder zur Objektinterpretation herangezogen werden. Das Ergebnis dieses Teilziels ist eine Beschreibung der Art Nummer 2, die heute häufig das Endergebnis von Bildanalysesystemen ist.

Endziel der Bildanalyse ist dann eine Beschreibung der Art Nummer 3, bei der neben die Analyse des Bildes die Erzeugung eines Plans für weitere Handlungen und die Erklärung, warum diese Handlungen ausgeführt werden müssen, tritt. Hierzu ist ausführliches Wissen über den Problembereich notwendig. Die zugehörigen Algorithmen, unter dem Begriff "semantische und pragmatische Analyse" zusammengefaßt, sind bisher nur in Ansätzen für sehr eng umgrenzte Anwendungsfälle bekannt, so daß eine derartige Beschreibung bisher noch nicht von einem Bildanalysesystem erzeugt werden kann. Gerade das nuklearmedizinische System des hiesigen Instituts zeigt aber, daß erste Schritte in diese Richtung bereits gegangen sind.

Neben die Blöcke Prozesse, Ergebnisse und Modell tritt als vierte Einheit des Bildanalysesystems die Kontrolle. Diese steuert den Fortgang der Analyse, ruft in Abhängigkeit von bereits erreichten Zwischenergebnissen Prozesse auf und beendet den Analyselauf, wenn ein akzeptables Endergebnis erreicht oder nicht mehr zu erreichen ist. Näheres kann dazu z.B. bei Nagao /NAG 82/ oder Niemann /NIE 83b/ nachgelesen werden. Daß die beiden Blöcke Erklärung und Lernen als eigenständige Module aufgefaßt werden, ist eine Entwicklung, die durch die immer größere Komplexität der Systeme bedingt wird. Die Erklärungskomponente soll, wie schon ihr Name sagt, die Ergebnisse in einer für den Benutzer geeigneten Art darstellen, erklären, woraus das System diese Ergebnisse schließt und gegebenenfalls Rückfragen zu den Resultaten beantworten können. Hierzu müssen die Abfolge der einzelnen Verarbeitungsschritte mitprotokolliert und geeignete Schlüsse daraus gezogen werden (vgl. /BUC 84/). Je umfangreicher das Modell

wird, desto aufwendiger ist dessen Akquisition. Deshalb möchte man das Modell möglichst automatisch lernen. Später will man auch während der Verarbeitung neue Strukturen in das Modell aufnehmen können. Für diese beiden Aufgaben steht der Block Lernen im Bildanalysesystem zur Verfügung.

Diese Arbeit befaßt sich mit dem Modell des Bildanalysesystems. Es werden Grundstrukturen aufgezeigt, die man für den Aufbau eines 3D-Modells benötigt. Dabei zeigt sich, daß zwei wesentliche Komponenten betrachtet werden müssen. Dies ist zum einen die Modellbildung für die Objekte des Problemkreises. Dazu müssen geeignete mathematische Modelle für die Eigenschaften der Objekte, die man innerhalb eines Bildanalysesystems braucht, gefunden werden. Zum anderen muß das Modell auf dem Rechner in adäquater Art und Weise gespeichert werden. Hierzu sind Formalismen zur Wissensrepräsentation bekannt (siehe Kapitel 4), aus denen einer, die relationale Datenbank, ausgewählt und angepaßt wurde.

2.2 Benötigtes Wissen

Um einschätzen zu können, welche Arten von Wissen im Modell zu berücksichtigen sind, soll hier versucht werden, dies kurz zu diskutieren. Dabei wollen wir nur Aspekte berücksichtigen, die direkten Einfluß auf Bildanalysesysteme und ihre Implementierung haben und keine allgemeine Diskussion des Begriffs Wissen führen. Wir nehmen dabei eine Unterteilung in problemabhängiges, problemunabhängiges und Metawissen vor.

2.2.1 Problemunabhängiges Wissen

Problemunabhängiges Wissen sind Daten und Fakten, die mehr oder minder für die gesamte Bildverarbeitung zutreffen und die sich nicht von Problemkreis zu Problemkreis ändern. Einige Beispiele sollen dies verdeutlichen:

- Die Abbildungseigenschaften bei der Bildaufnahme
- Beleuchtungsparameter und Schattenwurf
- Prinzipielle Fehler
- Reflexionseigenschaften von Oberflächen
- Das "Paradigma der Segmentierung": Abrupte Änderungen bezüglich eines Homogenitätskriteriums treten (fast) nur an Objektgrenzen auf.
- 3D-Transformationen und Projektionen

Da dieses Wissen allgemein verwendbar ist, wird es direkt in die Programme der Bildvorverarbeitung und der Segmentierung eingebracht und nicht extra abgespei-

chert. Oft werden dann auch die gesamten Programme als problemunabhängiges Wissen oder Methoden /SAG 85/ bezeichnet. Da dieses Wissen nicht explizit im Bildanalysesystem niedergelegt ist, spricht man auch von implizitem Wissen.

2.2.2 Problemabhängiges Wissen

Unter problemabhängigem Wissen wird das gesamte Wissen verstanden, das speziell den gerade betrachteten Problemkreis betrifft und das bei einem anderen Problemkreis ein anderes ist. Solches Wissen betrifft u.a.:

- Die Objekte des Problemkreises mit ihrem Aussehen, evt. Farbe, Textur usw.
- Die örtlichen Beziehungen der Objekte untereinander
- Abhängigkeiten, also die Frage, ob Objekte nur gemeinsam auftreten
- Die Variabilität der Objekte
- Die Frage, ob unbekannte Objekte auftreten können
- Evt. Wissen, das über "optisches" Wissen hinausgeht und für das Anstoßen einer Handlung notwendig ist (vgl. Beschreibung 3 aus Abschnitt 2.1)
- Wenn eine Bildfolge vorliegt: die Bewegungsmöglichkeiten der Objekte.

Man unterscheidet dabei zwischen prozeduralem und deklarativem Wissen. Bei prozeduralem Wissen werden Prozeduren oder Funktionen angegeben, die, abhängig von Eingabeparametern, spezielles Wissen berechnen. Deklaratives Wissen beschreibt einen Sachverhalt. Beispielsweise ist bei dem nuklearmedizinischen System /SAG 85/, das in Kapitel 1 kurz vorgestellt wurde, das Aussehen des Herzens in einem deklarativen Teil des Modells niedergelegt. Dagegen benötigt man für die Beschreibung von Bewegungsabnormalitäten eine Bewertung der Bild-zu-Bild-Änderung der Herzgröße. Diese Bewertung ist prozedural als Funktion gespeichert und wird erst während des Analyselaufs berechnet. Oft ist es möglich, ein Modell prozedural oder deklarativ aufzubauen. Dies haben Stein et al. /STI 86/ für die Werkstückerkennung durchgeführt. Sie haben zum Aufbau des Modells eines Wekstücks einmal einen prozeduralen Ansatz (erweitertes Übergangsnetzwert ATN) und einmal einen deklarativen Ansatz (Produktionensystem) verwendet. Die gestellte Aufgabe (Werkstückerkennung) wurde von beiden Ansätzen gelöst, aber jeweils nur mit einigen Einschränkungen. Nach Abwägung des Für und Wider kommen sie zu dem Schluß, daß beide Ansätze sich ergänzen sollten.

Problemabhängiges Wissen sollte innerhalb eines Bildanalysesystems vom problemunabhängigen Teil des Systems getrennt werden, um einen Wechsel des Problemkreises zu ermöglichen, ohne das gesamte System neu schreiben zu müssen. Deshalb gibt es in Bild 2.2 auch einen eigenen Block (Modell) dafür. Außerdem

müssen die Schnittstellen zu den anderen Blöcken so gestaltet sein, daß ein Wechsel des Problemkreises und damit des Modells nicht den Rest des Systems beeinflußt.

2.2.3 Metawissen

Metawissen bezeichnet das Wissen über das Wissen. Darin werden also alle im System explizit gespeicherten Informationen über das Wissen zusammengefaßt. Davis und Buchanan /BUC 84/ unterscheiden dabei vier Arten von Metawissen (siehe Bild 2.3).

Bild 2.3: Einteilung von Metawissen

Metawissen, das speziell die Darstellung des Wissens beschreibt, ist vor allem für die Wissensakquisition und das Lernen wichtig. Diese Verfahren können damit ohne Interaktion des Benutzers die Wissensbasis erweitern. Bei der Kontrolle kann durch explizite Angabe dieses Metawissens eine Änderung der Strukturierung des Modells abgekoppelt werden von der Änderung des Kontrollalgorithmus. Demgegenüber steht problemkreisspezifisches Metawissen, das Aussagen über die mögliche Bandbreite und hervorstechende Eigenschaften der Objekte und Beziehungen des Problemkreises macht, ohne dabei die Darstellung zu berücksichtigen. Darüberhinaus wird das Metawissen aufgrund der Quelle, aus der es kommt, unterschieden. Einerseits kann der Benutzer es definieren und in geeigneter Form im Rechner ablegen. Diese "herkömmliche" Methode entspricht heute noch vielfach dem Stand der Technik. In Zukunft werden Systeme entwickelt, die Metawissen bei der Akquisitions- oder in der Analysephase automatisch ableiten.

Das im weiteren vorgestellte Bildanalysemodellierungssystem zählt zum Metawissen des Systems, da ausgeführt wird, wie Objekte prinzipiell gegliedert werden und wie ein Modell aufgebaut wird. Davis und Buchanan schließen das Kapitel über Metawissen mit der Bemerkung, daß zukünftige Bildanalysesysteme

nicht mehr einfach mehr Wissen umfassen können, sondern daß dabei eine Grenze erreicht wird, wo Strategien für die effektive Nutzung dieses Wissens wichtig werden, wenn die Performanz eines Systems noch erhöht werden soll. Deshalb stellen "metawissensgestützte" System einen interessanten Forschungsgegenstand dar.

2.3 Theoretische Grundlegung des Modellbegriffs

Der Begriff des Modells ist in dieser Arbeit wie in der gesamten Informatik von wesentlicher Bedeutung. Deshalb wollen wir uns diesem Begriff etwas grundlegender nähern. Allerdings treten wir dabei nicht in die wissenschaftstheoretische Diskussion (siehe z. B. /CHL 86/) ein, sondern gründen uns auf die konstruktive Wissenschaftstheorie (speziell auf die "Erlanger Schule" /LOR 75/). Folgen wir Luft /LUF 84/, so stellen Modelle Schemata oder Muster von irgendwelchen benennbaren, nicht notwendigerweise physikalischen Gegenständen dar. Modelle bilden diese Gegenstände in vereinfachender Weise nach oder sie lassen diese Gegenstände durch ihr Vorhandensein erst erzeugen. Wesentlicher Punkt bei der Modellbildung ist also die Vereinfachung, d. h. man läßt gewisse Aspekte der Gegenstände wegfallen. Diese Vereinfachung ist dann vom Kontext, in dem die Modelle gebildet werden, abhängig.

Ziel dieser kurzen Untersuchung ist es, festzustellen, welche Art der Modellbildung in der Bildanalyse vorgenommen wird und welche Auswirkungen dies auf die Bildanalyse hat. In Anlehnung an Inhetveen /INH 86/, Kötter /KOE 86/ sowie Inhetveen und Luft /INH 83/ unterscheiden wir grundsätzlich folgende Modelle.

- Idealmodelle:
  Bei Idealmodellen wird angenommen, daß gewissse Parameter und Meßergebnisse nur bestimmte, ausgezeichnete Werte annehmen dürfen. Ein Beispiel dafür ist die Annahme, daß ein Punkt im dreidimensionalen Raum keine Ausdehnung besitzt. Man erhofft sich, dadurch erst einmal die Theorie aufzustellen und eventuell später noch geeignete Modifikationen vornehmen zu können. Idealisierungen sind nicht nur in der Physik, sondern auch in anderen Fachgebieten weit verbreitet.

- Veranschaulichungsmodelle:
  Diese werden sehr viel heuristischer aufgebaut als Idealmodelle, bei denen man genau angeben kann, welche Parameter und welche Wechselwirkungen idealisiert sind. Ausgehend von einer Beschreibung der zu modellierenden

Gegenstände werden Beziehungen der Gegenstände untereinander heuristisch dargestellt, so daß das aus dem gesamten Datenmaterial erzeugte Veranschaulichungsmodell keine offensichtlichen Fehler enthält. Ob die Veranschaulichungsmodelle tauglich sind, läßt sich erst anhand der empirischen Nachprüfung der abgeleiteten Gesetze beweisen.

- Aphairetische Modelle:
  Dies sind Modelle, die aus Normen oder Konstruktionsvorschriften abgeleitet werden. Dabei wird eine völlige Beachtung dieser Normen oder Vorschriften vorausgesetzt. Insofern erhält man eine ideale Beschreibung, wie ein Gegenstand sein soll, als Modell. Hier geht man bei der Modellbildung also von einer Regel aus, die sicher nicht vollständig der Wirklichkeit entspricht. Im Gegensatz dazu geht man bei Idealmodellen von einer realen Beschreibung aus, die dann idealisiert wird.

Nun ist die Frage zu klären, welche Art von Modell in der Bildanalyse verwendet wird und welche Auswirkungen dies auf die Performanz eines Bildanalysesystems hat. Betrachten wir einmal das computergestützte Konstruieren (CAD), bei dem aufgrund einer Konstruktionsvorschrift Modelle von Werkstücken oder Robotern erzeugt werden, so haben wir es dort mit einem rein aphairetischen Modell zu tun. Ohne Berücksichtigung realer Gegebenheiten werden die Modelle anhand von Rechen- und Konstruktionsvorschriften erzeugt und mit Hilfe von Beleuchtungs-, Schattierungs- und Abbildungsregeln auf dem Bildschirm ausgegeben. Als Ergebnis erhält man das "wunderschöne" Modell eines Werkstücks, dem man meist auf den ersten Blick ansieht, daß es eben kein reales Abbild ist.

In der Bildanalyse genügt dieses aphairetische Modell nicht, da man elementar von gemessenen Bildern, also realen Meßergebnissen abhängig ist. Auch Veranschaulichungsmodelle treffen nicht zu, da man sich nicht einen Sachverhalt durch ein Modell (eventuell eines anderen Sachverhalts) erläutern will. Insofern bietet es sich an, ein Idealmodell zu verwenden, in dem alle gemessenen Parameter auf ausgezeichnete Werte oder Wertebereiche begrenzt werden. Dies ist notwendig, um Interpretationen der im Bild gefundenen Gegenstände in der gewünschten Genauigkeit zu erreichen. Andererseits trifft man damit bei der Modellbildung eine Vorentscheidung über die Genauigkeit der Modellbildung, die man erhalten kann. Daraus sieht man, daß die Genauigkeit der Ergebnisse eines Bildanalyselaufs von der Genauigkeit der Modellbildung abhängt.

2.4 Geometrische 3D-Transformationen

Wir geben hier eine kurze Einführung in 3D-Transformationen, die später bei der Modellbildung gebraucht werden. Grundlage dieser Einführung sind /FOL 82/ und /NEM 79/. Da dadurch die Darstellung wesentlich vereinfacht wird, werden homogene Koordinaten verwendet.

Ein Punkt $P=(x,y,z)$ im 3D-Raum ergibt sich dann als
$P'=(x',y',z',w)=(w\cdot x,w\cdot y,w\cdot z,w)$ mit $w\neq 0$.
Man erhält also $P=(x,y,z)$ aus P' durch Division durch w wieder, $x=x'/w$, $y=y'/w$, $z=z'/w$.

Die Multiplikation eines homogenen Punktes $P'=(x',y',z',w)$ mit einem Skalar n ergibt einen homogenen Punkt $P''=(n\cdot x',n\cdot y',n\cdot z',nw)$. Beide Punkte P' und P'' repräsentieren aber den gleichen Punkt $P=(x'/w,y'/w,z'/w)$ im 3D-Raum. Damit sind P' und P'' im 3D-Raum gleich, was wir mit $P'\hat{=}P''$ bezeichnen, aber als homogene Punkte sind sie nicht gleich. Gleichheit als homogene Punkte bezeichnen wir mit =, also $Q'=Q''$ für $Q'=(3,4,5,1)$ und $Q''=(3,4,5,1)$

Für die Betrachtung der 3D-Transformationen setzen wir o.B.d.A. w=1 voraus. Mit den homogenen Koordinaten lassen sich alle 3D-Transformationen als (4x4)-Matrizen darstellen. Die Translation eines Punktes P' um T_x in x-Richtung, T_y in y-Richtung und T_z in z-Richtung wird durch die Matrix $T(T_x,T_y,T_z)$ beschrieben.

$$T(T_x,T_y,T_z)=\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ T_x & T_y & T_z & 1\end{pmatrix}$$

Damit ergibt sich $(x',y',z',1)\cdot T(T_x,T_y,T_z)=(x'+T_x,y'+T_y,z'+T_z,1)$

Die Skalierung um die Faktoren S_x,S_y und S_z wird mit der Matrix $S(S_x,S_y,S_z)$ ausgedrückt.

$$S(S_x,S_y,S_z)=\begin{pmatrix}S_x & 0 & 0 & 0\\ 0 & S_y & 0 & 0\\ 0 & 0 & S_z & 0\\ 0 & 0 & 0 & 1\end{pmatrix}$$

Oder ausgerechnet, $(x',y',z',1)\cdot S(S_x,S_y,S_z)=(S_xx',S_yy',S_zz',1)$.

Für die Rotation werden drei Matrizen $R_x(\theta)$,$R_y(\theta)$ und $R_z(\theta)$ gegeben, die die Rotation um die x-Achse, die y-Achse und die z-Achse jeweils um den Winkel beschreiben. Dabei gehen wir von einem Rechte-Hand-Koordinatensystem aus und geben die Rotation im Gegenuhrzeigersinn an, wenn man von einem Punkt auf der entsprechenden Achse zum Ursprung schaut (siehe Bild 2.4).

$$R_x(\theta)= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_y(\theta)= \begin{pmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_z(\theta)= \begin{pmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

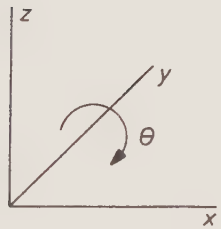


Bild 2.4: Koordinatensystem mit Rotation um die y-Achse

Nehmen wir z.B. den Einheitsvektor in x-Richtung (1001) und rotieren diesen um 90^o um die y-Achse, so erhalten wir: (1001) $R_y(90^o)$=

$$(1001) \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = (0\ 0\ -1\ 1)$$

Das Ergebnis dieser Rotation ist der Einheitsvektor in negativer z-Richtung, wie man auch an Bild 2.4 nachprüfen kann.

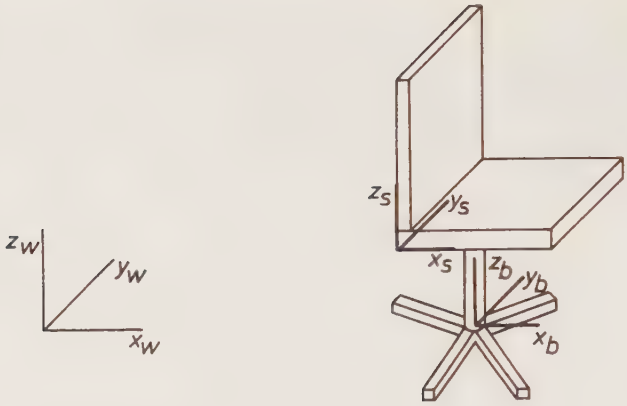


Bild 2.5: Ein Drehstuhl

An dem Beispiel eines Drehstuhls (Bild 2.5) soll demonstriert werden, wie diese Transformationen bei der Modellbildung verwendet werden können. In diesem Bild gibt es 3 Koordinatensysteme (KOS). Das Welt-KOS (x_w,y_w,z_w) dient als Hauptreferenz für die gesamte Szene. Assoziiert mit dem Sitz des Stuhles ist das Stuhl-KOS(x_s,y_s,z_s). Als dittes KOS tritt das KOS des drehenden Zylinders auf (x_b,y_b,z_b). Sei ein Punkt P_i gemessen im System i, also z.B. P_s gemessen im Stuhl-KOS. Sei weiterhin T_{bs} die Transformation, die den Ort des Zylinder-KOS relativ zum Stuhl-KOS angibt. Damit ergibt sich

$$P_s=P_bT_{bs}.$$

Dreht sich nun der Stuhl, dann entspricht dies einer Rotation um die z_b-Achse. Dies ist eine Transformation innerhalb des Zylinder-KOS. Wir erhalten dann

$$P_s=P_bT_{bb}T_{bs}.$$

Betrachtet man diese Drehung aber im Stuhl-KOS, so entspricht ihr eine Transformation T_{ss} innerhalb des Stuhl-KOS, die dann nach T_{bs} ausgeführt werden muß, also

$$P_s=P_bT_{bs}T_{ss}.$$

Man sieht damit, daß die Reihenfolge der Transformation von der geeigneten Beschreibung abhängig gemacht werden kann. Einen Punkt P_s im Stuhl-KOS kann man dann auch ins Welt-KOS transformieren.

$$P_w=P_sT_{sw}.$$

Setzt man nun für P_s noch die ursprüngliche Transformation $P_s=P_bT_{bs}$ ein, so erhält man:

$$P_w = P_b T_{bs} T_{sw}.$$

Den dabei verwendeten Transformationen T_{ij} liegt jeweils die Überführung eines KOS in ein übergeordnetes zugrunde, was wir später in ähnlicher Weise wie in Bild 2.5 nutzen werden.

3. Ansätze zur Modellbildung für visuelle Daten

Dieses Kapitel soll verschiedene Ansätze zur Modellbildung für visuelle Daten zusammentragen. Es geht also um Verfahren, die bei Programmsystemen eingesetzt werden, die mit optischer Aufnahme und/oder Wiedergabe arbeiten. Dabei wird mehr Gewicht auf die dreidimensionale gegenüber der zweidimensionalen Modellbildung gelegt. Zu Beginn werden einige Kriterien erläutert, die für die Modellbildung von Bedeutung sind. Dann folgt die Vorstellung einiger aus der Literatur oder dem industriellem Einsatz bekannter Verfahren, bevor abschließend die drei wichtigsten Anwendungsgebiete, CAD, Computeranimation und Bildanalyse kurz vorgestellt werden.

3.1 Grundsätzliche Überlegungen zur Modellbildung für visuelle Daten

Um während eines Bildanalyselaufs eine symbolische Beschreibung des Bildes zu erzeugen, benötigt das System ein Modell des Problemkreises. Bei der Antwort auf die Frage nach Anforderungen und Einschränkungen für die Objekte, die zu modellieren sind, folgen wir Requicha /REQ 80/. Objekte sind feste Körper, die Teilmengen des dreidimensonalen Euklid'schen Raums E^3 darstellen und folgende Eigenschaften haben müssen:

- Fest: die Form des Körpers muß fest sein und darf sich nicht ändern in Abhängigkeit von Position oder Orientierung.
- Dreidimensional homogen: Ein Körper muß ein Inneres haben, und zu der Oberfläche des Körpers dürfen keine "baumelnden" Kanten und Flächen gehören (siehe Bild 3.1).
- Endlich: Ein Körper darf nur einen endlichen Teilraum umfassen.

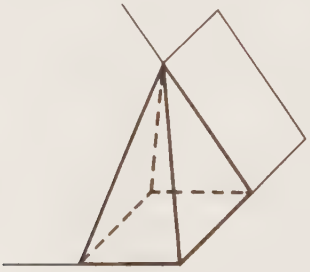


Bild 3.1: Pyramide mit "baumelnden" Kanten und Flächen

- Abgeschlossen unter Bewegung und Booleschen Operationen: Bei der Bewegung im Raum und bei der Veränderung eines festen Körpers muß wieder ein fester Körper entstehen. Die Booleschen Operationen können dies nicht gewährleisten, so daß sogenannte regularisierte Boolesche Operationen ($\cup^*, \cap^*, -^*$) eingeführt werden, die wie die "normalen" Operationen definiert sind und zusätzlich "baumelnde" Kanten und Flächen eliminieren (siehe Bild 3.2).
- Endlich beschreibbar: Es muß möglich sein, eine Repräsentation im Computer, die notwendigerweise endlich ist, zu generieren.
- Bestimmt durch die Oberfläche: Die Oberfläche des Körpers muß unzweideutig festlegen, was das Innere des Körpers ist und deshalb den Körper umgeben.

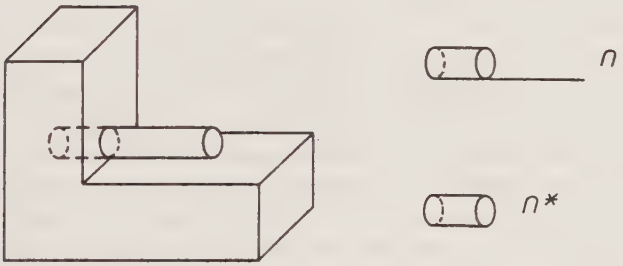


Bild 3.2: Regularisierter Durchschnitt

Diese Definition der festen Körper berücksichtigt nur die Geometrie und die Oberflächeneigenschaften. Requicha bezeichnet sie als Modelle M. Vorausgegangen ist hier der Schritt von den Objekten der realen Welt zu diesen festen Körpern, den Requicha nicht erwähnt. Auf diesen Schritt wurde in Abschnitt 2.3 dieser Arbeit eingegangen. Nun müssen diese festen Körper M in syntaktisch korrekte Repräsentationen R gebracht werden. Hierzu dient ein Repräsentationsschema s. Es gilt also: $M \overset{s}{\supseteq} R$. Dieses Repräsentationsschema s ist eine Relation, da es prinzipiell einzelne feste Körper geben kann, für die keine Repräsentation existiert, ebenso wie es möglich ist, daß syntaktisch korrekte Repräsentationen auftreten, die keinen festen Körper als Entsprechung haben. Im weiteren Verlauf der Arbeit werden wir wieder von Modellbildung in dem gängigen Sinne reden, die den gesamten Prozeß von den Objekten des Problemkreises bis zu den Repräsentationen R umfaßt. Für die ersten beiden Abschnitte dieses Kapitels scheint es aber günstiger, sich auf die mathematisch besser faßbare Relation s, $M \overset{s}{\supseteq} R$, zu beschränken.

Sei $V \subseteq R$ die Menge der Repräsentationen, für die mindestens ein fester Körper existiert, also der Bildbereich von R. Jedes $r \in V$ wird gültig genannt. Eine gültige Repräsentation r ist unzweideutig oder vollständig, wenn sie genau einem festen Körper $m \in M$ entspricht, also $s^{-1}(r) = m$. Eine gültige Repräsentation r ist eindeutig, wenn die festen Körper, die auf r abgebildet werden, auf keine weitere Repräsentation abgebildet werden, also $s(s^{-1}(r)) = r$. Ein Repräsentationsschema s heißt dann vollständig oder eindeutig, wenn alle gültigen Repräsentationen vollständig bzw. eindeutig sind. Diese Definitionen bilden nun die Grundlage für die Beschreibung von Kriterien und Eigenschaften von Repräsentationsschemata, die im nächsten Abschnitt aufgeführt werden.

3.2 Kriterien für die Modellbildung

Für die Beurteilung von Repräsentationsschemata gibt es verschiedene Kriterien, die aufzeigen, welche Eigenschaften ein ideales Repräsentationsschema haben soll. Wir wollen diese Kriterien in drei Gruppen einteilen, von denen die ersten beiden Gruppen aus dem computerunterstützten Konstruieren (CAD) stammen. Obwohl das Ziel - Erzeugung eines Modells auf dem Computer - hier anders ist als bei der Bildanalyse, sind diese Kriterien doch so grundlegend bei der Modellbildung, daß sie für beide Anwendungsbereiche zutreffen. Dabei trennen wir diese Kriterien in Anlehnung an Requicha /REQ 80/ und Brown /BRW 81/ in formale und informelle Kriterien. Die dritte Gruppe umfaßt Kriterien, die vor allem in der Bildanalyse von Bedeutung sind, dort die Interpretation eines Bildes mit einem Modell unterstützen und von Brady /BRD 83/, Binford /BIN 82/, Marr /MAR 82a/ sowie Heß und Göbel /HES 85/ stammen.

3.2.1 Formale Kriterien für die geometrische Modellbildung

Diese vier Kriterien beziehen sich direkt auf die in Abschnitt 3.1 formalisierte Relation $M \overset{\rightarrow}{\underset{s}{\leftarrow}} R$. Sie machen Aussagen über das Repräsentationsschema s und seine Auswirkungen auf feste Körper M und die Repräsentationen R.

- Definitionsbereich: Der Definitionsbereich eines Repräsentationsschemas s umfaßt alle die festen Körper, die durch s beschrieben werden können. Diese Menge ist von Repräsentationsschema zu Repräsentationsschema durchaus unterschiedlich und kann deshalb ein bestimmtes Repräsentationsschema für einen Problemkreis untauglich erscheinen lassen.

- Gültigkeit: Entsprechend der Definition in 3.1 ist ein Repräsentationsschema

s gültig, wenn alle r ∈ R gültig sind, d.h. zu jedem r ∈ R mindestens ein fester Körper m ∈ M existiert. Es muß einfach möglich sein, dieses Kriterium abzuprüfen. Denn es muß vermieden werden, daß eine Repräsentation erzeugt wird, die keinem festen Körper entspricht. Wenn beim geometrischen Modellieren in Zukunft vermehrt Repräsentationen automatisch erzeugt werden, dann sollte das verwendete Repräsentationsschema einen automatischen Gültigkeitstest ermöglichen.

- Vollständigkeit: Ein Repräsentationsschema s ist vollständig (siehe Abschnitt 3.1), wenn es zu jeder Repräsentation r ∈ R genau einen festen Körper gibt. Dieses Kriterium erleichtert und ermöglicht in vielen Fällen eine vereinfachte Berechnung von Eigenschaften, da eindeutige Berechnungsvorschriften gegeben werden können. Trotzdem gibt es einige Repräsentationsschemata, die dieses Kriterium nicht erfüllen und dennoch in verschiedenen Anwendungsfällen tauglich und aus anderen Gründen vorteilhaft sind.

- Eindeutigkeit: Eindeutigkeit, d.h., daß es zu jedem festen Körper, der repräsentiert wird, genau eine Repräsentation gibt, erleichtert wesentlich den Vergleich von zwei Repräsentationen, da nicht zwei unterschiedliche Repräsentationen einem festen Körper entsprechen können.

Repräsentationsschemata, die sowohl vollständig als auch eindeutig sind, sind theoretisch am besten, da sie eine eineindeutige Abbildung liefern. Damit kann man Gleichheit von Objekten durch die Gleichheit der Repräsentationen abtesten, wobei diese syntaktisch aufgebaut sind, so daß eine Syntax-Prüfung genügt. Ein einfaches Beispiel sind die natürlichen Zahlen, bei denen es klar ist, daß "258" und "66" unterschiedliche Zahlen darstellen.

3.2.2 Informelle Kriterien für die geometrische Modellbildung

Hier werden drei Kriterien aufgeführt, die nicht in direktem Zusammenhang mit der mathematischen Formulierung des Problems stehen, sondern allgemeiner Art sind. Sie sind deshalb auch mehr qualitativ als quantitativ, d.h. für diese Kriterien ist kein absolutes Maß anzugeben. Trotzdem müssen sie bei der geometrischen Modellbildung beachtet werden, wenn ein handhabbares und anwendbares System entstehen soll.

- Prägnanz: Die einzelnen Repräsentationen sollen möglichst klein sein und möglichst wenig Redundanz enthalten. Damit verringern sich Speicherplatz-

bedarf und Übertragungszeiten. Daneben sollte eine geringe Redundanz die Menge der Zweideutigkeiten in den Repräsentationen erniedrigen. Andererseits kann, wie man aus der Codierungstheorie weiß, gerade Redundanz zur Erkennung und Korrektur von Fehlern beitragen.

- Einfachheit bei der Erzeugung: Für die Benutzung des Modellierers ist es wichtig, daß die Generierung von Repräsentationen einfach ist, vor allem, wenn diese durch den Benutzer erfolgt. Dabei sind prägnante Repräsentationen sicher von Vorteil, da der Benutzer nicht eine Unmenge von Daten und Nebeneffekten beachten muß. Je schwieriger die Erzeugung ist, desto wichtiger ist es auch, eine automatische Gültigkeitsprüfung in das System zu integrieren.

- Effizienz bei Anwendungen: Als letztes Kriterium in dieser Gruppe wird die Effizienz genannt. Da geometrische Modellbildung als Hilfsmittel in Anwendungssystemen fungiert, muß sie so ausgerichtet sein, daß die dort vorkommenden Algorithmen und Verfahren effizient implementiert werden können. Dies kann auch bedeuten, daß ein zweideutiges Repräsentationsschema in diesem Sinn besser ist als ein unzweideutiges, das eigentlich nach der Theorie vorzuziehen ist.

3.2.3 Kriterien für die Bildanalyse

In der dritten Gruppe sind Kriterien zusammengefaßt, die spezielle Eigenschaften des Modellierers betreffen, die vor allem in der Bildananlyse von Belang sind. Dabei laufen einige dieser Kriterien konträr zu den vorher definierten. Dort muß man dann immer einen von der Anwendung diktierten Kompromiß finden.

- Lokale Unterstützung: Es muß möglich sein, Teile einer Repräsentation lokal zu berechnen, ohne die gesamte Repräsentation zu erzeugen. Dies ist in der Bildanalyse wichtig, da man aufgrund von Verdeckungen oder Aufnahmefehlern nicht immer mit dem vollständigen Abbild eines Objekts rechnen kann.

- Integration: Hat man die Möglichkeit, lokale Teile von Repräsentationen zu berechnen, so müssen damit auch Repräsentationen zusammengefaßt werden können. Dabei werden auch solche Teile eliminiert, die zwar lokal zutreffen, aber nicht in den globalen Kontext passen.

- Fortpflanzung: Wenn man mehrere lokale Teile oder lokale Parameter hat, die in irgendeiner Weise zusammenpassen, so sollten diese auch miteinander in

Verbindung gebracht werden können. Zur Verdeutlichung sei folgendes kleine Beispiel erläutert: Die Drehung des Drehstuhls aus Bild 2.4 wird lokal um die z_b-Achse erzeugt. Die Drehung der Sitzfläche sollte dann auch durch eine Drehung um die entsprechende Achse in der Sitzfläche beschrieben werden.

- Oberflächenattribute: Die Oberflächen der repräsentierten Körper beinhalten für die Bildanalyse die meiste Information. Deshalb muß ein Repräsentationsschema hier die Definition von vielerlei Information zulassen und einen schnellen Zugriff erlauben, der unabhängig von einem langwierigen Aufbau einer vollständigen Repräsentation ist. Da dies bei geometrischen Modellierern aus dem CAD nicht der Fall ist, generieren Bolles et al. /BOL 83/ aus dem CAD-Modell für die Bildanalyse ein sog. Eigenschaftennetzwerk, in dem allein die Oberflächenattribute für einen schnellen Zugriff gespeichert sind.

- Zugänglichkeit: Während des Bildanalyselaufs muß ein Zusammenhang zwischen den Repräsentationen, die dreidimensonal sein können, und dem zweidimensionalen Bild hergestellt werden. Je mehr das Repräsentationsschema dies unterstützt, desto einfacher und schneller kann die Bildanalyse ablaufen.

- Objektklassenbildung: In vielen Problembereichen der Bildanalyse können von vorne herein unbekannte Objekte auftreten, da nicht für alle möglichen Objekte eine Repräsentation abgespeichert werden kann. Hierfür werden dann generische Repräsentationen oder, wie wir es nennen, Objektklassen erzeugt. Innerhalb der Definition von Objektklassen können auch Fehler der Aufnahme oder der Bildvorverarbeitung in gewissem Umfang abgefangen werden.

Diese Kriterien zeigen, daß für die Bildanalyse geeignete Repräsentationsschemata neben den aus dem CAD bekannten Eigenschaften noch einige andere Merkmale haben müssen. Deshalb ist es nicht möglich, einen geometrischen Modellierer aus dem CAD direkt in einem Bildanalysesystem einzusetzen. Da es aber sinnvoll ist, CAD-Daten in der industriellen Bildanalyse zu nutzen, werden zwei Vorgehensweisen untersucht. Bolles et al. /BOL 83/ sowie Bhanu und Henderson /BHA 85,HEN 85/ erweitern das CAD-Modell um die für die Bildanalyse notwendigen Komponenten, während Castore und Crawford /CAT 84/ aus dem CAD-Modell für die Bildanalyse geeignete Informationen extrahieren.

3.3 Ansätze zur Modellbildung

Dieser Abschnitt stellt einige Ansätze zur Modellbildung vor, wobei vor allem die Verfahren berücksichtigt sind, die in der Bildanalyse Anwendung finden. Aus der Menge der Ansätze werden die gängigsten ausgewählt, so daß einige Verfahren, wie z.B. Skelettierung /LOB 80/, Linien- und Flächennachbarschaftsgraphen /PAV 82/, die SLS-Darstellung ("smoothed local symmetries") /BRD 83/ und die Darstellung mit überlappenden Kugeln /ROU 79/, ausgelassen werden. Die vorgestellten Verfahren werden jeweils am Beispiel des in Bild 3.3 gezeigten U-förmigen Klotzes erläutert.

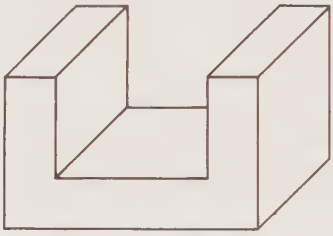


Bild 3.3: Beispielobjekt

Um uns dem allgemeinen Sprachgebrauch anzupassen, sprechen wir im weiteren wieder von Modellbildung, wenn wir den gesamten Prozeß ansprechen, von den Objekten des Problemkreises bis zu den in den Abschnitten 3.1 und 3.2 Repräsentationen genannten $r \in R$, die in Zukunft Objektmodelle oder, wenn es aus dem Zusammenhang klar ist, Modelle heißen. Die gesamte Menge der Repräsentationen heißt dann das Modell, wobei in dieses auch die Beziehungen der Objekte untereinander eingehen.

3.3.1 Übersicht

Eine erste Trennung wird zwischen zweidimensionalen und dreidimensionalen Verfahren zur Modellbildung vorgenommen. Bisher finden in der Bildanalyse die 2D-Verfahren weitere Verbreitung. Dies liegt vor allem an der Tatsache, daß damit im Modell Merkmale und Eigenschaften gespeichert sind, die direkt aus dem Bild zu gewinnen sind. Man erspart sich also die Umsetzung eines 3D-Modells auf 2D-Projektionen davon oder die Gewinnung von 3D-Merkmalen und 3D-Eigenschaften aus dem Bild. Bei den 2D-Verfahren zur Modellbildung kann man unterscheiden

zwischen Verfahren, die lediglich Kennzahlen verwenden, konturlinienorientierten und flächenorientierten Methoden.

Dreidimensionale Modellbildung hat den Vorteil der Standpunktunabhängigkeit. Jede Ansicht und jede 2D-Projektion eines Objekts ist aus dem Objektmodell abzuleiten. Dies wird benötigt, wenn die Orientierung eines Objekts bei der Analyse nicht bekannt ist. Die oft verwendete Möglichkeit, ein Objekt durch mehrere 2D-Ansichten zu modellieren, ist nur durchführbar, wenn die Anzahl dieser Ansichten, wie es in der industriellen Bildanalyse oft der Fall sein kann, sehr begrenzt ist. Ein weiterer Vorteil der 3D-Modelle liegt darin, daß die Objekte aus den Modellen wieder rekonstruiert werden können. Grundsätzlich gibt es 3D-Modelle auf der Basis von Oberflächen und von Volumen. In der Bildanalyse realisierte Oberflächenmodellierer verwenden meist planare Flächen, mit denen gekrümmte Flächen approximiert werden. Modellierer, die vor allem die Synthese von Objektmodellen in den Vordergrund stellen, wie beim CAD und bei der Computeranimation, umfassen meist auch Definitionsmöglichkeiten für Flächen höherer Ordnung (siehe z.B. /BAS 86/, /FOL 82/). Bei den Volumenmodellen werden vor allem drei Ansätze verfolgt. In Analogie zu Bildpunkten ("picture elements, pixels") wird eine Aufteilung in würfelartige Volumenelemente ("voxels") vorgenommen und damit Objektmodelle gebildet. Als zweites gibt es rotations- und translationssymmetrische Körper und Ableitungen davon, die verallgemeinerten Zylinder, die auch in dieser Arbeit untersucht werden. In vielen CAD-Systemen findet sich die konstruktive Geometrie fester Körper ("constructive solid geometry, CSG"), die, aufbauend auf primitiven Körpern, mit Booleschen Operationen die Erzeugung von Objekten definiert.

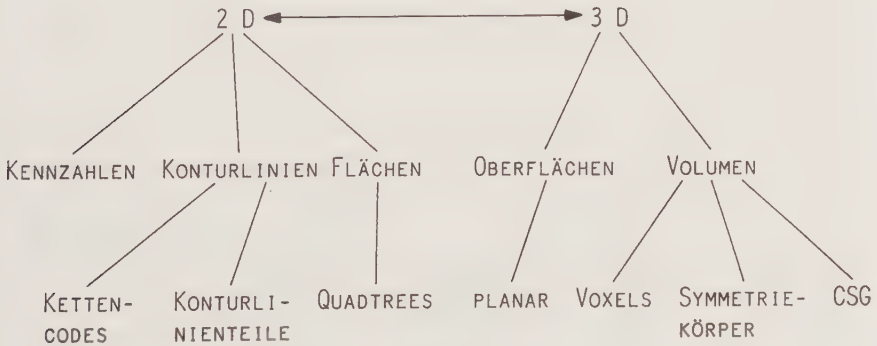


Bild 3.4: Ansätze zur Modellbildung - Übersicht

3.3.2 Ansätze zur 2D-Modellbildung

Bei der 2D-Modellbildung werden Ansichten, d.h. Projektionen, des Objekts modelliert. Wenn die Orientierung des Objekts, die man auf dem Bild erwartet, unbekannt ist, muß man mit entsprechend vielen Ansichten arbeiten. Deswegen sind diese Verfahren nur für spezielle Problemkreise geeignet. Bild 3.5 zeigt eine Ansicht des Objekts aus Bild 3.3 und dient als Grundlage für die Beispiele in Abschnitt 3.3.2. Das Objekt ist in einem Bild mit 16 x 16 Bildpunkten aufgenommen. Das Bild ist binärisiert und insofern ideal, da weder bei der Quantisierung noch bei der Binärisierung Fehler aufgetreten sind.

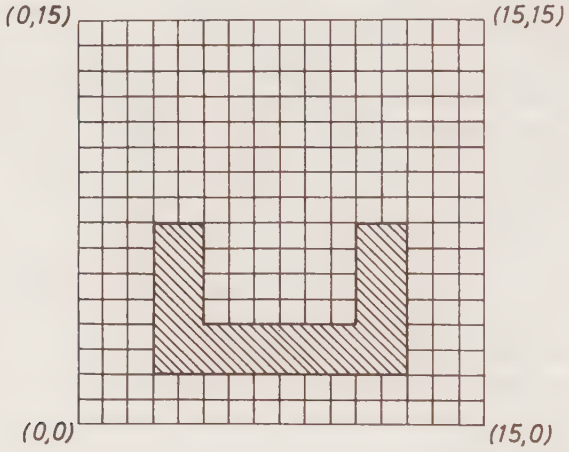


Bild 3.5: Eine Ansicht des Beispielobjekts

Kennzahlen:
Dies sind meist heuristisch gewonnene Merkmale (siehe z.B. /AGI 80/ und /NIE 83a/), die für eine begrenzte Anzahl von Objekten innerhalb eines Problemkreises unterscheidende Wirkung haben. Einige Beispiele sind:

- Das umschreibende Rechteck MBR: Dabei wird das kleinste Rechteck angegeben, das die Ansicht des Objekts beinhaltet.
- Die Länge des Umfangs L: Diese Angabe erfolgt in Bildpunkten.
- Die Fläche des Objekts A: Diese Angabe erfolgt in Bildpunkten.
- Der Formfaktor $F = L^2/A$.
- Der Winkel w zwischen den beiden Richtungen des größten und des kleinsten

Abstands der Kontur vom Schwerpunkt.
- Die Anzahl der Löcher innerhalb des Objekts.

Für die Ansicht in Bild 3.5 errechnen sich folgende Werte:
- MBR = (1,2), (8,13). Diese Angabe erfolgt durch die beiden Eckpunkte, da angenommen wird, daß das umschreibende Rechteck parallel zur x- und y-Achse liegt.
- L = 36
- A = 36
- $F = L^2/A = 36$
- $w = 39^o$
- Anzahl der Löcher = 0.

Die Interpretation eines Bildes mit einem solchen Modell ist einfach, aber es gibt drei gravierende Nachteile. Aus dem Objektmodell ist keine Rekonstruktion des Objekts möglich. Da die Kennzahlen jeweils die gesamte Ansicht bewerten, müssen die Bilder die Ansicht vollständig sowie ohne Verdeckungen zeigen und es dürfen keine Fehler bei der Berechnung auftreten. Darüberhinaus ist der Definitionsbereich dieser Art von Modellbildung durch die starke Reduktion der Daten sehr eingeschränkt.

(2,3) 0 0 0 0 0 0 0 0 0 1 1 1 1 1 2 3 3 3 3
2 2 2 2 2 2 2 1 1 1 1 2 3 3 3 3 3

(2,3) 0 0 0 0 0 0 0 0 0 2 2 2 2 2 4 6 6 6
5 4 4 4 4 4 3 2 2 2 4 6 6 6 6 6

Bild 3.6: Kettencode von Bild 3.5

Auf der Basis von Konturlinien:
Das einfachste Verfahren zur Modellbildung mit Konturlinien ist die Kettencodierung /FRE 74/. Entsprechend der vorgegebenen Codierung (0-3 oder 0-7) wird die Kontur der Ansicht eines Objekts verfolgt und als Modell aufgenommen. Das Modell für die Ansicht in Bild 3.5 ist in Bild 3.6 aufgeschrieben. Hat man nun ein Objekt, $A = a_1 a_2 \ldots a_n$, im Kettencode modelliert, so kann man zur Erkennung

die Kettenkreuzkorrelation $\Phi_{ab}(j)$ einsetzen. Sei dazu ein Bild, eine Fläche B mit Kettencode $b_1 b_2 \ldots b_n$ gefunden, dann gilt

$$\Phi_{ab}(j) = \frac{1}{n} \sum_{i=1}^{n} \cos((a_i - b_{i+j})\pi/4) \text{ für die Codierung 0-7 und}$$

$$\Phi_{ab}(j) = \frac{1}{n} \sum_{i=1}^{n} \cos((a_i - b_{i+j})\pi/2) \text{ für die Codierung 0-3.}$$

Da bei diesem Verfahren ziemliche Einschränkungen gelten, hat es keine weitere Verbreitung gefunden. Kettencodes finden hauptsächlich als sehr kompakte Codierung für Flächen in Bildern Verwendung.

Eine weitere häufig verwendete Möglichkeit, Konturlinien zu codieren, ist die Angabe in Geradenstückchen, Kreisen und Kreisbögen (/SHI 78, PER 78/). Yachida und Tsuji /YAC 77/ unterwerfen das Objekt einer Polarkoordinatentransformation und machen dann folgende Angaben: Ein Polygonzug wird durch eine Liste $P_k(r_k,\theta_k)$ der Eckpunkte, ein Kreis durch den Mittelpunkt $P_c(r,\theta)$ sowie seinen Radius und ein eventuell approximierter Kreisbogen durch Mittelpunkt $P_c(r,\theta)$, Radius r und die Winkel der beiden Enden des Bogen beschrieben. Dies alles wird in eine verkettete Liste aufgenommen, die dann das Modell einer Objektansicht darstellt. Mittels einer geeigneten Bewertungsfunktion kann der Erkennungsprozeß ablaufen. Der Vorteil liegt darin, daß auch eine unvollständige Kontur noch zu geeigneten Ergebnissen führen kann. Allerdings verbieten sich durch die Verwendung der Polarkoordinatentransformation Verdeckungen oder nur teilweise vorhandene Objekte.

Auf der Basis von Flächen:
Zur Darstellung von Flächen sind Quadbäume ("quadtrees" /SAM 84/) eine häufig benutzte Datenstruktur. Zu ihrer Erläuterung gehen wir von einem quadratischen Bild aus, das in vier gleich große Quadrate zerlegt wird. Diese werden mit NW, NO, SW und SO bezeichnet (man denke dabei an die Himmelsrichtungen in einer Landkarte). Jedes Quadrat kann entweder vollständig vom Objekt erfüllt sein, dann wird es mit "schwarz" ■ bezeichnet. Es kann in diesem Quadrat überhaupt kein Teil des Objekts liegen, was zur Beschriftung "weiß" □ führt, oder das Quadrat ist gemischt und wird "grau" o genannt. Ist das Quadrat grau, so wird es wieder in vier Quadrate aufgeteilt. Geht man von einem quadratischen Bild mit der Seitenlänge 2^n aus, so läßt sich dieser Prozeß bis auf Bildpunktebene fortsetzen, wo dann eine Beschriftung mit schwarz oder weiß erfolgen muß. Damit erhält man einen Quadbaum, wie er in Bild 3.7 für das Objekt aus Bild 3.5 aufgeführt ist. Zu Quadbäumen gibt es eine Menge von Varianten, die jeweils

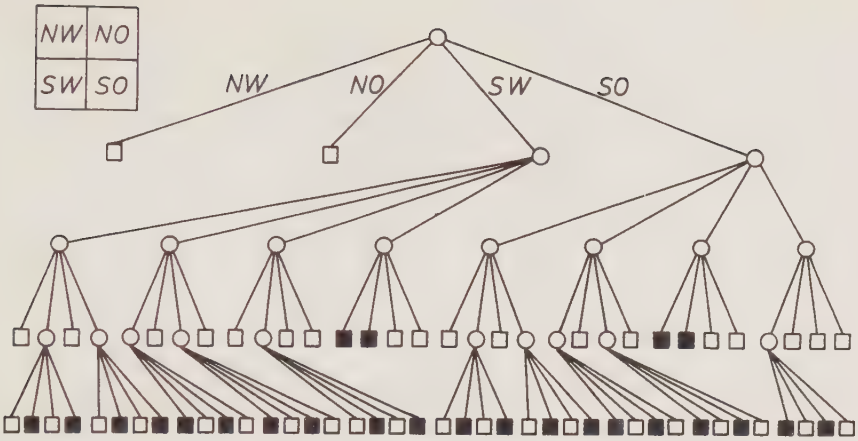


Bild 3.7: Ein Quadbaum für Bild 3.5

eine hervorstechende Eigenschaft aufweisen (siehe /SAM 84/). Da Quadbäume aber vor allem bei der Synthese von Bildern verwendet werden, verweisen wir hier auf die Literatur (siehe z.B. /TAN 80/ und /PAV 82/).

3.3.3 Ansätze zur 3D-Modellbildung

Der schon erwähnte Vorteil der 3D-Modellbildung liegt in der Standpunktunabhängigkeit. Damit fallen die Einschränkungen auf Ansichten u.ä. weg. Jedes Objektmodell erhält hierfür ein lokales Koordinatensystem zugeordnet, um auch lageunabhängige Objektmodelle zu haben. Die Bewertung der einzelnen Modellierer, die anhand der in Abschnitt 3.2 aufgestellten Kriterien vorgenommen wird, stützt sich u.a. auf Brown /BRW 81/ und Göbel /GOB 85/.

Die einfachste Form der 3D-Modellbildung sind Drahtgittermodelle, die nur die Eckpunkte und die Kanten eines Objekts modellieren. Bild 3.8 zeigt ein Objekt, das aus den Eckpunkten A = (0,0,0), B = (5,0,0), C = (5,2,0), D = (0,2,0), E = (1,0,1), F = (4,0,1), G = (4,2,1), H = (1,2,1), I = (0,0,3), K = (1,0,3), L = (1,2,3), M = (0,2,3), N = (4,0,3), O = (5,0,3), P = (5,2,3) und Q = (4,2,3) sowie den ungerichteten Kanten AB, BC, CD, DA, EF, FG, GH, HE, AI, IM MD, ML, LK, KE, LH, FN, NQ, QG, QP, PO, ON, OB und PC aufgebaut ist. Drahtgittermodelle können nur Objekte mit geraden Kanten darstellen. Da keine Oberflächenattribute definierbar sind, und die Darstellung zweideutig ist, sind Drahtgittermodelle für die Bildanalyse unbrauchbar /BES 85/. Allerdings ist ihre Verarbeitung so

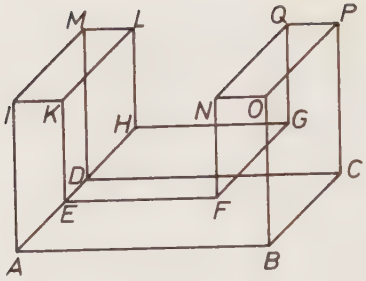


Bild 3.8: Drahtgitterdarstellung von Bild 3.3

einfach, daß sie bei Computergraphikanwendungen immer noch benutzt werden.

Oberflächen:
Ähnlich den Konturlinien bei einer 2D-Ansicht, werden 3D-Objekte von einer 3D-Oberfläche begrenzt, die den Körper umfassen soll. Zur Modellbildung wird diese Oberfläche in Teile, sog. Flächen ("faces"), zerlegt. Eine Fläche soll dabei eine Teilmenge der Oberfläche sein und keine "baumelnden" Kanten oder isolierte Punkte enthalten. Die Vereinigung aller Flächen soll die gesamte Oberfläche des Objekts ergeben. Eine Zerlegung der Oberfläche unseres Beispielobjekts in zehn Flächen zeigt Bild 3.9. Diese Zerlegung ist einsichtig, was dadurch zustande kommt, daß es sich um planare Flächen handelt. Ist die Oberfläche des Objekts aber gekrümmt, so wird es schwierig, den Begriff der Fläche sinnvoll zu fassen. Man denke dabei nur an eine Teilfläche einer Kugeloberfläche. Hier empfiehlt es sich, zur Modellbildung dann keine planaren Flächen mehr zu verwenden. Ballard und Brown /BAL 82/ diskutieren einige Ansätze mit gekrümmten Flächen, die mit Splines oder durch Funktionen der Kugeloberfläche definiert werden.

Bei der Modellbildung mit planaren Flächen wird meist angenommen, daß die Flächen durch gerade Kanten begrenzt werden. Diese Kanten können dann durch ihre zwei Endpunkte im dreidimensionalen Raum, eventuell in lokalen Koordinaten, beschrieben werden. Auf diese Weise entsteht ein Netz zur Beschreibung der Geometrie eines Objekts, woraus in Bild 3.10 ein Ausschnitt dargestellt ist. Erwähnt sei noch eine weitere Vereinfachung, die durch die Aufteilung der Flächen in Dreiecke, die sog. Dreieckszerlegung ("triangulation"), zustande kommt, bei der jede Fläche dann nur noch durch genau drei Kanten begrenzt ist. Dieser Vereinfachung stehen bei der Dreieckszerlegung zwei Nachteile gegenüber. Einerseits werden zusätzliche, nicht sichtbare Kanten erzeugt, und andererseits

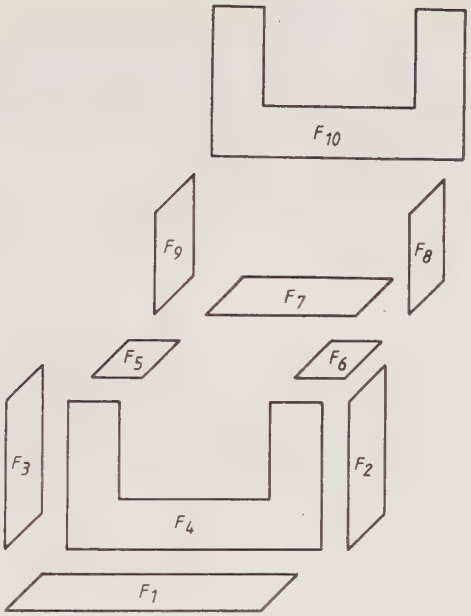


Bild 3.9: Oberflächenzerlegung des Beispielobjekts

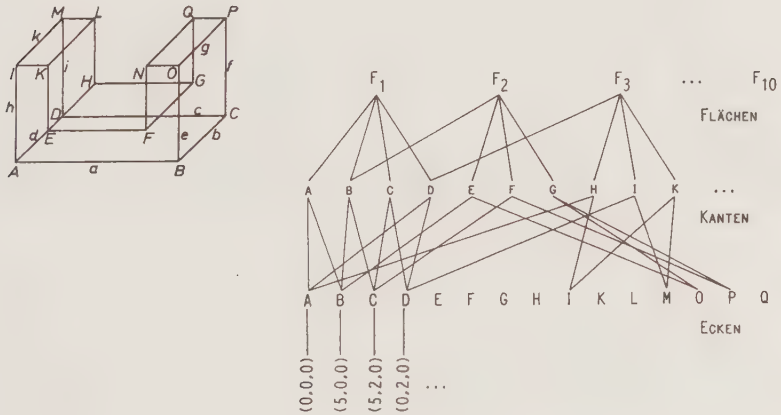


Bild 3.10: Teil des Netzes zur Oberflächenzerlegung

ist die Dreieckszerlegung nicht unzweideutig. Der Definitionsbereich der Oberflächendarstellung ist abhängig von der Form der Flächen, die verwendet wird, kann aber prinzipiell alle Objekte umfassen. Sehr aufwendig ist der Nachweis der Gültigkeit eines Objektmodells. Ist ein Objektmodell gültig, so ist es auch vollständig, wenn die Flächen unzweideutig dargestellt sind. Eine Objektklassenbildung ist bei der Oberflächendarstellung möglich. Bei der lokalen Unterstützung, der Definition von Oberflächenattributen und bei der Zugänglichkeit weist diese Darstellung besondere Stärken auf, so daß sie durchaus berechtigt in vielen Systemen eingebaut ist.

Voxel:

Bei der Modellbildung mit Volumenelementen (Voxels) denke man sich den Raum in drei Dimensionen gerastert. Darin liegt dann in der gleichen Rasterung das gewünschte Objekt (siehe Bild 3.11). Wenn man ein Modell mit einigen Details in Voxels ausdrücken will, benötigt man eine große Datenmenge. Deshalb verwendet man sog. Octbäume ("octtrees" oder "octrees"), um eine geeignete Datenreduktion

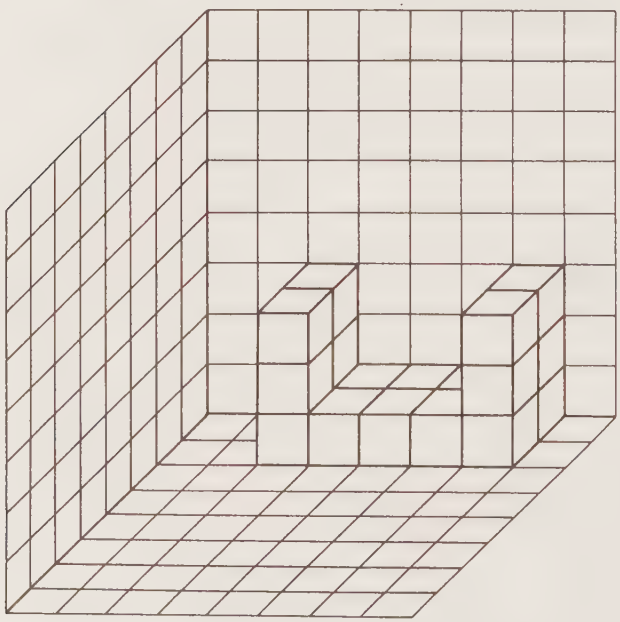


Bild 3.11: Voxeldarstellung des Beispielobjektes

bei der Abspeicherung zu erzielen. Analog den Quadbäumen sollte die Rasterung des Raumes eine Kantenlänge von 2^n Voxels haben. Bei jedem Schritt zur Erzeugung eines Octbaumes wird der Raum in acht Teile A,B,C,D,E,F,G und H zerlegt, die vom Objekt ausgefüllt ■ , ohne Objekt □ oder gemischt o sein können. Bei den gemischten Volumen wird dieser Prozeß fortgesetzt, bis kein gemischtes Volumen mehr vorhanden ist. Der Octbaum zu Bild 3.11 hat das in Bild 3.12 gezeigte Aussehen. Weitere Möglichkeiten, Voxels geeignet zu codieren, sind bei Srihari /SRI 81/ diskutiert. Die Modellbildung mit Voxels ist vollständig und

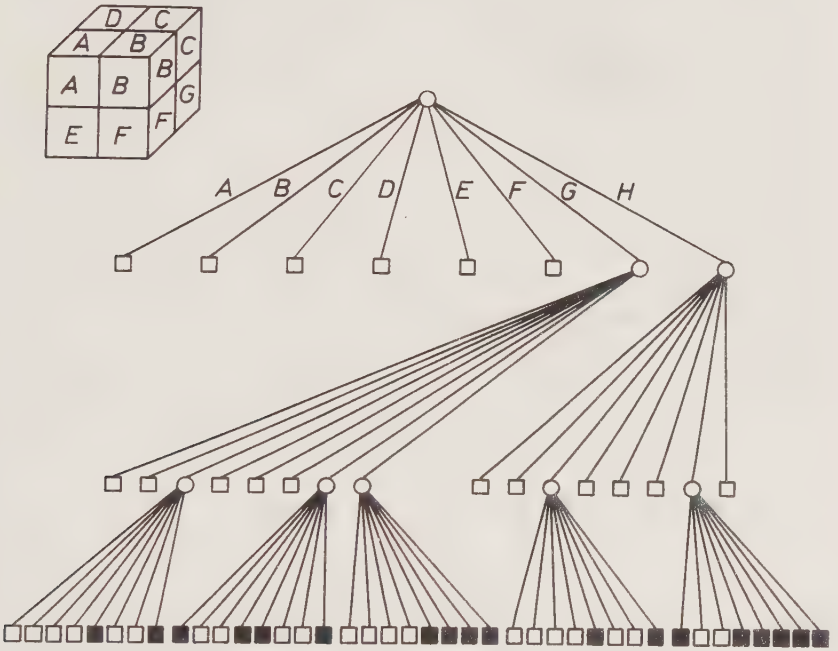


Bild 3.12: Ein Octbaum zu Bild 3.11

eindeutig. Auch der Nachweis der Gültigkeit ist relativ einfach. Die Verarbeitung ist allerdings wegen der hohen Datenmenge sehr aufwendig. Deshalb ist die Zugänglichkeit sehr gering und eine Klassenbildung nur mit hohem Aufwand möglich. In der Bildanalyse finden daher Voxels kaum Verwendung in der Modellbildung.

Symmetriekörper:

Es gibt viele Objekte, für die sich ein translations- oder rotationssymmetrisches Objektmodell bilden läßt. Hierzu wird eine Fläche entlang einer Achse oder um eine Achse bewegt, wodurch ein dreidimensionaler Körper entsteht (siehe Bild 3.13). Als Erweiterung hierzu wurden verallgemeinerte Zylinder (VZ) einge-

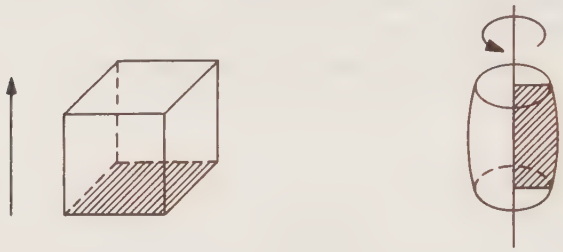


Bild 3.13: Zwei Symmetriekörper

führt. Diese sind in der allgemeinen Form durch eine dreidimensionale Kurve im Raum, genannt Achse, definiert, zu der normal (im 90^{o}-Winkel) an jedem Punkt der Kurve eine Fläche (die sog. "cross section") existiert (siehe Bild 3.14). Diese Definition ist allerdings nicht handhabbar. Deshalb werden bei Implementierungen immer wesentliche Einschränkungen vorgenommen. Agin und Binford /AGI 76/ lassen als Achse eine Gerade und als Flächen Kreise zu, wodurch der Definitionsbereich des Modellierers eingeschränkt wird. Zur Modellierung des mensch-

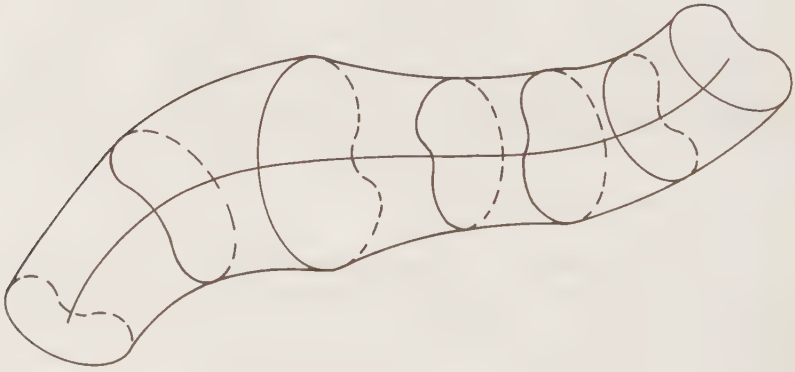


Bild 3.14: Ein verallgemeinerter Zylinder

lichen Bauches verwendet Shani /SHA 80/ einen VZ, dessen Achse als kubische Spline definiert ist. Die Flächen werden mit einer geschlossenen Kurve beschrieben, die wiederum ein kubischer Spline ist. Wegen der Verarbeitung im Rechner gibt es nur eine endliche Anzahl von Flächen. Trotzdem ist dieser sehr allgemeine Ansatz für ein Modell, das aus vielen Objektmodellen besteht, zu aufwendig. Zuletzt sei noch die Arbeit von Brooks /BRO 81/ erwähnt, der einen VZ als Translationskörper durch Achse, Grundfläche und Veränderungsregel ("sweeping rule") definiert. Dabei ist die Achse eine gerade, nicht parametrisierte Linie, die Grundfläche ein Rechteck oder ein Kreis, und die Veränderungsregel gibt als lineare Funktion die Länge der Diagonale bzw. des Durchmessers in Abhängigkeit vom Schnittpunkt mit der Achse an. Um noch weitere Objektmodelle aufzubauen, können mehrere VZ mit einem Vereinigungsoperator zusammengefaßt werden. Das in Kapitel 5 vorgestellte Bildanalysemodellierungssystem verwendet ein ähnliches Verfahren zur geometrischen Modellbildung. Andere Arten der Zusammensetzung (über Berührungspunkte und mit gemeinsamen Achsen) diskutiert Agin /AGI 81/ unter Verwendung der VZ mit der Definition von Brooks /BRO 81/. Für reine Symmetriekörper gilt, daß sie gültig und unzweideutig sind, wenn die 2D-Flächen dies sind. Prinzipiell sind sie nicht eindeutig (der Würfel aus Bild 3.13 hat mehrere Symmetrieachsen) und ihr Defintionsbereich ist ziemlich beschränkt. Dieser ist bei VZ bedeutend erweitert, jedoch ist die Gültigkeit je nach Definition schwieriger nachzuweisen, ebenso wie die Zugänglichkeit bei anderen Modellierern einfacher ist. Dagegen ist die lokale Unterstützung und die Klassenbildung gerade bei zusammengesetzten VZ gut zu lösen.

CSG:

Bei der konstruktiven Geometrie fester Körper ("constuctive solid geometry", CSG) hat man einfache Primelemente, die mit Booleschen Operatoren verknüpft werden. Um die Defintion der Objektmodelle zu vervollständigen, muß jeweils angegeben werden, wie die Primelemente oder ganze Teilbäume zueinander bewegt werden müssen. Bild 3.15 zeigt einen CSG-Baum für das Objekt aus Bild 3.3, der durch die Differenz zweier Quader zustandekommt. Die Erzeugung von CSG-Bäumen wird in folgender Syntax festgelegt:

<CSG-Baum> ::= <Primelement>/
<CSG-Baum><Boolescher Operator><CSG-Baum>/
BEWEGE<CSG-Baum>MIT<BEW.PARAMETER>

Als Boolesche Operatoren werden vor allem Vereinigung , Durchschnitt und Differenz - ausgewählt, wobei wegen der Forderung nach Abgeschlossenheit (siehe Abschnitt 3.1) die regularisierten Operatoren $\cup^*$, $\cap^*$ und $-^*$ verwendet werden. Primelemente werden als die Schnittmenge von abgeschlossenen Halbräumen, die

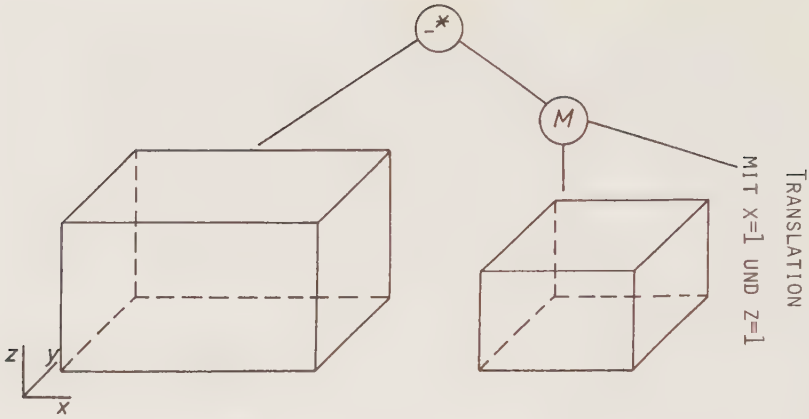


Bild 3.15: Ein CSG-Baum des Beispielobjekts

durch eine "einigermaßen vernünftige" Funktion F(x,y,z) mit F(x,y,z) ≥ 0 beschrieben werden, definiert. Für den großen Quader aus Bild 3.15 gilt:

$$x \geq 0 \quad y \geq 0 \quad z \geq 0$$
$$-x+5 \geq 0 \quad -y+2 \geq 0 \quad -z+3 \geq 0.$$

Entsprechend gilt für den kleinen Quader:

$$x \geq 0 \quad y \geq 0 \quad z \geq 0$$
$$-x+3 \geq 0 \quad -y+2 \geq 0 \quad -z+2 \geq 0.$$

Die CSG-Darstellung ist nicht eindeutig, wie aus Bild 3.16 (ein zweiter CSG-Baum für dasselbe Objekt) zu sehen ist, aber sie hat einen großen Definitionsbereich und ist vollständig. Die Gültigkeit hängt bei der Verwendung von regularisierten Operatoren nur an den Primelementen, ist also einfach nachzuweisen. Klassenbildung ist möglich, auch wenn die Zugänglichkeit nicht ganz so einfach ist. Da CSG-Modelle sehr einfach zu handhaben sind, werden sie im CAD-System oft zur Eingabe benutzt. Allerdings werden die Objektmodelle intern in die Oberflächendarstellung umgewandelt.

3.4 Anwendungsbeispiele

In diesem Abschnitt wird anhand der drei Anwendungsgebiete CAD, Computeranimation und Bildanalysesysteme kurz dargelegt, welche Verfahren zur Modellbildung dort zum Einsatz kommen.

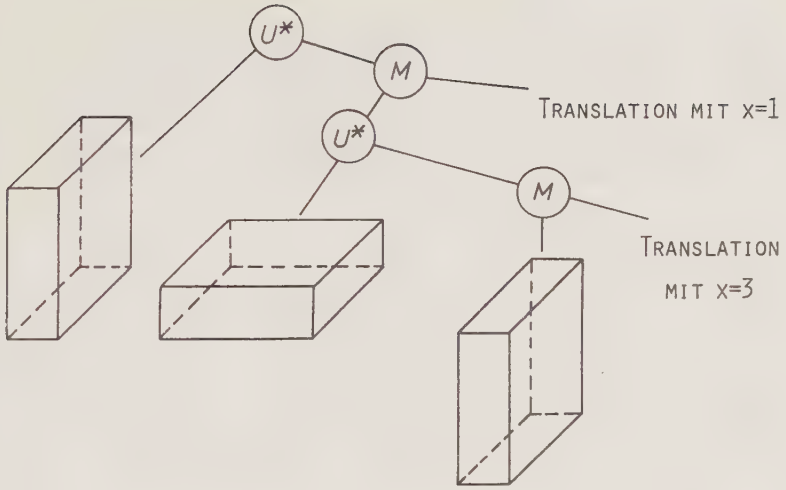


Bild 3.16: Ein weiterer CSG-Baum des Beispielobjekts

3.4.1 CAD-Systeme

Bei CAD-Systemen geht der Trend zu 32-Bit-Arbeitsstationen mit hochauflösendem grafischem Bildschirm und dem Betriebssystem UNIX /SIM 85/. Bei den Verfahren zur Modellbildung beginnt die Integration von 2D-Zeichensystemen und 3D-Konstruktionssystemen, die bisher noch getrennt sind, so daß man entweder das eine oder das andere System kaufen muß. Deshalb betrachten wir im weiteren nur reine 3D-Konstruktionssysteme. Die geometrischen Modellierer in diesen Systemen arbeiten meist mit mehreren verschiedenen Verfahren zur Modellbildung. Dies soll die wesentlichen Vorteile der einzelnen Methoden ausnützen. Deshalb wird intern die Oberflächendarstellung verwendet, da diese Darstellung für die Visualisierung der Modelle am besten geeignet ist. Zur Eingabe wird dann in vielen Systemen ein CSG-Verfahren angeboten, da der Operator mit diesem Formalismus leichter umgehen kann als mit der vom Aufbau her etwas unanschaulichen Oberflächendarstellung. Tabelle 3.1, zusammengestellt aus /REQ 82/ und /REQ 83b/ führt die geometrischen Modellierer für einige bekannte CAD-Systeme auf.

Als ein Beispiel soll noch kurz das System EUCLID von Matra Datavision erwähnt werden (vgl. /MAA 83/, /SPU 84/, /COO 85/). EUCLID ist als Softwarepaket oder als Gesamtsystem mit einem VAX-Rechner, einem speziellen Displayprozessor und dem grafischen Bildschirm zu erhalten. Die Generierung von Objektmodellen bei

System	Hersteller	Grundsystem	Eingabe
CADD	MCAuto	Oberflächen	Symmetriek.
CATIA	IBM	Oberflächen	
COMPAC/UNISCAD	TU Berlin/Sperry	Oberflächen	CSG,Symm.
CATSOFT	Catronix	CSG	
EUCLID	Matra Datavision	Oberflächen	CSG,Symm.
ICEM SOLID M.	CDC	CSG	CSG
MEDUSA	Prime	Oberflächen	CSG,Symm.
PADL 1+2	U. Rochester	CSG	CSG
ROMULUS	Shapedata	Oberflächen	CSG,Symm.,EOP*
SOLIDESIGN	Computervision	Oberflächen	
SYNTHAVISION	MAGI	CSG	CSG
TIPS-1	CAM-I, Hokkaido U.	CSG	CSG

*EOP = Euler Operationen

Tabelle 3.1: Verschiedene CAD-Systeme

EUCLID beginnt der Operator mit Basiselementen. Dies können Punkte, Geraden, Kreisbögen, polygonale Linien und Kurven n.ten Grades (bis n=8) sein, die benutzt werden können, um Flächen auf vielerlei Arten zu definieren. Volumenelemente können Translations- oder Rotationskörper, Polyeder, Strangvolumen (Querschnittsfläche und Mittellinie) oder Volumen, die durch Teile von Regelflächen oder komplexen Flächen begrenzt werden. Damit hat man nun Grundvolumen definiert, auf die jetzt Transformationen und die mengentheoretischen Operationen Vereinigung, Differenz (hier Subtraktion genannt), Durchschnitt und beliebige Schnitte (siehe Bild 3.17) angewendet werden können. So erhält man Objektmodelle, sog. Figuren, die weiterverarbeitet werden können. Hierfür ste-

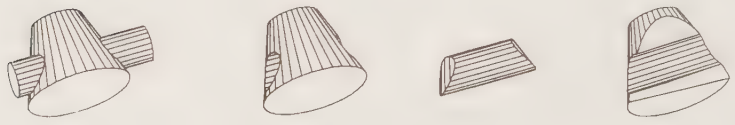


Bild 3.17: Operatoren aus EUCLID

hen mannigfaltige Darstellungsmöglichkeiten, wie Projektionen aus verschiedenen Blickwinkeln, Vergrößerung, Ausschnitte, Entfernung verdeckter Linien, Bemaßung und die Einblendung von Schrift, zur Verfügung. Daneben gibt es noch ca. 20 Anwenderpakete, aus allen möglichen Industriesparten, die die generierten Objekte entsprechend den dort gestellten Anforderungen weiterverarbeiten.

3.4.2 Computeranimation

Die Anforderungen in der Computeranimation sind anders als in der Bildanalyse. Wichtig ist hier ein möglichst realistisches Abbild, das synthetisch aufgebaut wird. Deshalb wird sehr viel Gewicht auf die optischen Gesetze von Belichtung, Schattierung und dreidimensionalem Abbild gelegt. Entsprechend diesen Forderungen sind die Objekte meist durch dreidimensionale Oberflächen modelliert, wobei eine beliebte Vorgehensweise die Approximation durch Polyederflächen mit einer nachträglichen Glättung ist. Verschiedenartige Computerbilder und ihre Entstehung zeigt z.B. Deken /DEK 83/.

Als Beispiel wollen wir noch das System Mira-3D etwas näher betrachten /THA 83, THA 85/. Dort definiert man Figuren in PASCAL-ähnlichen Prozeduren und zeichnet sie mit Attributen versehen auf ein Ausgabegerät. Es gibt eine Menge von dreidimensionalen Standardfiguren:

- Typ: Linie
- planare Figuren: Dreieck, Quadrat, Kreis, Ellipse,...
- Typ: Ebene mit drei Vektoren definiert
- Typ: Box mit vier Vektoren definiert
- Typ: Kugel approximiert durch Polygone
- die regulären Polyeder: Tetraeder, Würfel, Octaeder, Dodekaeder, Ikosaeder
- Oberflächentypen: Zylinder, zylinderartig, parametrisierte Oberfläche, Bezier-Oberfläche, Coons-Oberfläche, B-Spline-Oberfläche.

Aus diesen Figuren werden dann Objekte, wie die Giraffe in Bild 3.18, erzeugt.

3.4.3 Bildanalysesysteme

Einen Überblick über verschiedene Bildanalysesysteme findet sich bei Binford /BIN 82/ oder Bunke /BUN 85/. Dieser Abschnitt gründet sich auf eine eigene Untersuchung /HES 84/. Dabei wurden 15 zufällig gewählte Systeme speziell im Hinblick auf die Modellbildung untersucht. Als einziges Auswahlkriterium galt eine Verarbeitung von Einzelbildern. Dazu wurde das schon erwähnte nuklearmedizinische System aufgenommen. Allerdings wird auch hier nur der Teil des

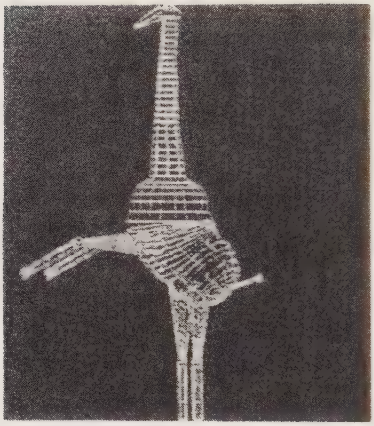


Bild 3.18: Bild einer Giraffe aus /THA 83/

Modells berücksichtigt, der sich auf ein einzelnes Bild bezieht und die Modellierung der Herzmuskelbewegung nicht betrachtet. Alle Angaben sind aus der zitierten Literatur extrahiert.

Grundbausteine der Modelle sind die Grundsymbole. Diese sind die kleinste Einheit, die noch explizit im Rechner dargestellt wird. Da auf die Grundsymbole häufig zugegriffen wird, sollten sie nicht zu kompliziert aufgebaut sein. Es bieten sich als Grundsymbole natürlich Strukturen und Elemente an, die mittels Bildverarbeitungsalgorithmen direkt aus dem Bild gewonnen werden können. So bilden Kreissegmente, Kreise, Geradenstückchen und Flächen in vielen Bildanalysesystemen die Grundsymbole. Diese Grundsymbole sind alle zweidimensional. Dagegen stehen einige Systeme, die bereits als Grundsymbole verallgemeinerte Zylinder in ihrem Modell eingebaut haben. Bild 3.19 zeigt auf, welche Grundsym-

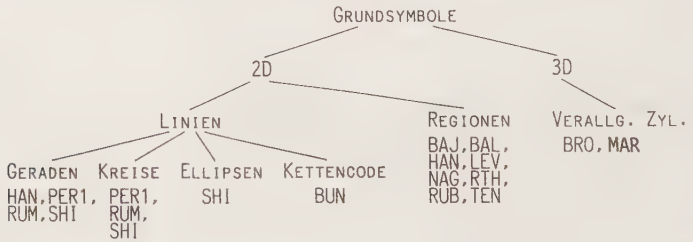


Bild 3.19: Grundsymbole

Tabelle 3.2: Objektmodelle in Bildanalysesystemen

Name	Objekt(Bsp.)	2-D	3-D	Standpunkt abh	Standpunkt unabh	oertliche Relation zu anderen Objekten	Teile des Obj. moegl.	Beschreibung Spekt	Beschreibung Form	Darstellung
BAJ	Strassen	x		x		durch Strategienetz	nein	x	x	als Prozedur
BAL	Schiffe			x		durch Kanten			x	semantisches Netz
BRO	Flugzeuge		x		x	im Bedingungsgraph	x	x	x	Graph
BUN	Herz	x		x		Left of, Right of	part of		x	semantisches Netz
HAN	Haeuser,Baeume	x		x		in sog. Schemas	x	x	Flaechen	semantisches Netz
LEV	Haus	x		x		durch Relationen	x	x		relationale Datenbank
MAR			x			nicht ausgefuehrt	x		x	
NAG	Haeuser	x		x		nein	nein	x	x	als Prozeduren
PER1	Industrieteile	x		x		nein	nein		x	Menge von Eigenschaften
PER2	Platinen	x		x		nein	x	x		Menge von Eigenschaften
RTH	Zebrastreifen	x		x		mittels Produktion		x	Kontext	Produktionen
RUB	Hochhaus	x		x		mittels Netzwerk	nein	x		Merkmalsvektor
RUM	Industrieteile	x		x		nein	nein		x	Menge von Eigenschaften
SHI	Telefon	x		x		nein	nein	wenig	Konturen	Eigenschaften
TEN	Stuhl	x		x		sog. Kontext	x	x	Kontext	semantisches Netz

Tabelle 3.3: Gesamtmodell in Bildanalysesystemen

Name	Szene / Daten	Anzahl der Objekte	mehrere Szenen	verschiedene Blickwinkel	Darstellung	Hierarchie	Literatur
BAJ	Luftbilder	nur Strassen	nein	nein	Eigenschaftsliste	nein	/BAJ 74/
BAL	Luftbilder(Hafen)	Schiffe,Docks	p. m.	nein	Bedingungsgraph	im Bedingungsgraph	/BAL 78/
BRO	Luftbilder (Flugplatz)	nur Flugzeuge	p. m.	ja	Graphen	in den Graphen	/BRO 81/
BUN	Bildfolge (Herz)	eines (Herz)	nein	nein	semantisches Netz	im semant. Netz	/BUN 83/
HAN	Vorstadtszene	beliebig	p. m.	nein	semantisches Netz	7 Schichten	/HAN 78/
LEV	Vorstadtszene	beliebig	p. m.		relationale Datenbank	Relation Containement	/LEV 78/
MAR			p. m.	ja		ja	/MAR 82/
NAG	Luftbilder	beliebig aus Felder Haeuser,Wald usw	nein	nein	Eigenschaftstafel	nein	/NAG 80/
PER1	Werkteile	einzelne Objekte	nein	nein	Merkmalsvektor	nein	/PER 78/
PER2	Platine	versch. Platinen	nein	nein	Eigenschaftsliste	Bauteil-Platine	/PER 84/
RTH	Luftbilder	beliebig	nein	nein	Produktionensystem	visuell,konzeptuell	/RTH 81/
RUB	Bilder(Pittsburgh)	beliebig	p. m.	ja	Graph(Locus Search)	nein	/RUB 78/
RUM	Werkteile	mehrere Werkteile	nein	nein	Eigenschaftsliste	nein	/RUM 84/
SHI	Bueroszene	beliebig	nein	nein	Eigenschaftstafel	nein	/SHI 78/
TEN	Bueroraum		nein	nein	semantisches Netz	im semant. Netz	/TEN 74/

p. m. prinzipiell moeglich

bole in den 15 untersuchten Systemen verwendet werden. Die Referenzen auf die jeweiligen Systeme sind in Tabelle 3.3 angegeben.

Bei der Untersuchung der Modellbildung wurde zwischen Objektmodellen einzelner Objekte und dem Gesamtmodell unterschieden. Die Ergebnisse sind in den Tabellen 3.2 und 3.3 zusammengefaßt. Die Vor- und Nachteile von 3D-Modellbildung wurden schon oben diskutiert. Zusätzlich sei hier aber noch auf einige Anwendungsgebiete, wie z.B. Satellitenaufnahmen, verwiesen, bei denen sich schon wegen der Aufnahmebedingungen 2D-Modellbildung anbietet. In Tabelle 3.2 ist jeweils ein Beispielobjekt, das in der angegebenen Literatur beschrieben wird, herausgegriffen und erläutert, ob es zweidimensional oder dreidimensional modelliert ist, ob das Objektmodell unabhängig vom Betrachterstandpunkt modelliert ist, ob und wie örtliche Beziehungen zu anderen Objekten realisiert werden können, ob das Objekt in Teile zerlegt werden kann und welche Oberflächenattribute zur Beschreibung des Objektmodells dienen. Der Eintrag Darstellung bezieht sich ebenso wie in Tabelle 3.3 auf den verwendeten Repräsentationsformalismus und wird in Kapitel 4 kommentiert.

Beim Gesamtmodell interessiert die Anzahl der verschiedenen Objekte oder Objektklassen, die modelliert werden können und ob es möglich ist, die Strukturen des Modellierers auch für andere Problemkreise zu verwenden (Eintrag: mehrere Szenen) oder ob Wissen und Metawissen in dem System doch so weit verquickt sind, daß man eigentlich ein neues System schreiben muß, wenn man den Problemkreis ändern will. "Verschiedene Blickwinkel" weist auf die Möglichkeit verschiedener Kamerastandpunkte hin. Dies ist eng verknüpft mit der Frage nach 2D- oder 3D-Modellbildung. Eine interessante Variante hat hier Rubin implementiert, der zwar zweidimensionale Objektmodelle benützt, aber zwischen diesen variable örtliche Beziehungen zuläßt, um verschiedene Blickwinkel zu modellieren. In der letzten Spalte spiegeln sich noch verschiedene Möglichkeiten wider, innerhalb des Gesamtmodells eine Hierarchisierung zuzulassen.

4. Ansätze zur Wissensrepräsentation

Im letzten Kapitel wurden verschiedene Ansätze zur Modellbildung visueller Datenverarbeitung diskutiert. Hat man eines dieser Verfahren ausgewählt, so stellt sich als nächster Schritt die Frage nach einer geeigneten Speicherung des Modells auf dem Rechner. Hierzu sind etliche Formalismen bekannt, die in diesem Kapitel kurz erläutert werden sollen. Der erste Abschnitt dieses Kapitels versucht Grundlagen und Ziele aufzuzeigen. Im zweiten Abschnitt werden in aller gebotenen Kürze die gängigen Verfahren vorgestellt, und der dritte Abschnitt beschließt dieses Kapitel mit einer Abwägung der Vor- und Nachteile dieser Methoden im Hinblick auf die Bildanalyse.

4.1 Grundlagen und Ziele der Wissensrepräsentation

Der Begriff Wissensrepräsentation wird heute vor allem im Umfeld der künstlichen Intelligenz (KI) gebraucht. Wissensrepräsentation gilt als eines der Grundlagenprobleme der KI, und entsprechend umfassend, aber auch noch umstritten sind Forschung und Veröffentlichungen auf diesem Gebiet. Ein Verfahren zur Wissensrepräsentation muß Datenstrukturen und Prozeduren zur Interpretation bereitstellen. Diese Prozeduren müssen in irgendeiner Form intelligente Ableitungen zulassen, so daß es möglich ist, abgespeicherte Daten nicht nur wiederzugewinnen, sondern auch mit Hilfe dieser Daten Schlüsse zu ziehen.

Wir wollen, von der Bildanalyse kommend, diese Abgrenzung etwas weniger strikt ziehen und auch Formalismen zur Wissensrepräsentation zulassen, die zwar Strukturen bereitstellen, um auf die zugehörigen Datenstrukturen zuzugreifen, die aber darüber hinaus das Ziehen von Schlüssen aus diesen Daten nicht unterstützen. Dadurch wird natürlich das projektierte Gesamtziel eines Bildanalysesystems, nämlich aus den interpretierten Daten Aktionen abzuleiten (Nii 86a), erschwert, aber eingedenk der Tatsache, wie weit man heute noch von diesem Ziele entfernt ist, erscheint es sinnvoll, Wissensrepräsentationsformalismen im Hinblick auf die Abspeicherung des Modells zu betrachten und deshalb die weitergehende Forderung nach dem daraus resultierenden Schließen fallenzulassen. So kristallisiert sich folgende Zielsetzung für die Formalismen zur Wissensdarstellung in der Bildanalyse heraus: Ein solcher Formalismus muß Datenstrukturen für die Speicherung der Objektmodelle und der Beziehungen der Objektmodelle untereinander sowie Strukturen für geeigneten und schnellen Zugriff auf diese Datenstrukturen bereitstellen.

Um verschiedene Verfahren zur Wissensrepräsentation bewerten und vergleichen zu können, kann man sich einige Fragen stellen:

- Welches Aussehen haben die Datenstrukturen?
- Welche Möglichkeiten des Zugriffs gibt es und wie schnell kann man dabei wahlfrei an ein bestimmtes Datum gelangen?
- Wie schwierig ist das Hinzufügen oder Ändern von Daten? Welche Abhängigkeiten innerhalb der Struktur kann man dabei beachten?
- Wie einfach ist der Umgang mit diesem Formalismus?
- Wird darstellungsspezifisches Metawissen explizit gespeichert?

Der Wissensdarstellung kommt innerhalb eines modellgestützten Bildanalysesystems eine wesentliche Bedeutung zu. Da während eines Analyselaufes viele Male auf das Modell zugegriffen wird, hängt die Effizienz des Systems u.a. von der Geschwindigkeit dieser Zugriffe ab.

4.2 Verfahren zur Wissensrepräsentation

In diesem Abschnitt werden die gängigen Verfahren kurz vorgestellt. Eine detaillierte Darlegung der Formalismen kann man unter besonderer Berücksichtigung der Implikationen aus der künstlichen Intelligenz z.B. bei Barr und Feigenbaum /BAR 81/ oder McCalla und Cercone /CAL 83/ nachlesen. Aus dem Blickwinkel der Bildanalyse wird die Wissensrepräsentation z.B. bei Ballard und Brown /BAL 82/ oder bei Niemann /NIE 81/ betrachtet. Als Beispiel werden wir bei einigen Verfahren auf die Syntax eines CSG-Baums (siehe Abschnitt 3.3.3) und auf einen einfachen Vertreter davon (siehe Bild 3.15) zurückgreifen. Allerdings werden dies immer Spielbeispiele bleiben, die vor allem das Prinzip verdeutlichen sollen.

Logik:

Prädikatenlogik erster Ordnung wurde von Mathematikern und Philosophen vor allem im 19. und beginnenden 20. Jahrhundert entwickelt. Sie arbeitet mit Prädikaten, also Aussagen der Form ist-rot oder ist-größer-als, die immer den Wert wahr oder falsch haben. Eingesetzt können in diese Prädikate nicht nur Individuen, sondern auch Variable, die mit den Quantoren es existiert $\exists$ oder für alle $\forall$ versehen sind. Daneben gibt es Funktionen, die als Ergebnis ein Objekt liefern können, wie z.B. Sohn-von oder die Addition. Prädikate und Funktionen dürfen allerdings nicht mit Quantoren spezifiziert sein. Wesentlich ist nun, daß es eine Menge Regeln, die sog. Inferenzregeln, gibt, mit denen aus wahren Tatsachen weitere Tatsachen abgeleitet werden können, die auch wahr sind. Umgekehrt kann man auch von einer gegebenen Tatsache abprüfen, ob sie

wahr oder falsch ist. Der große Vorteil gegenüber anderen Repräsentationsverfahren liegt in dieser Eindeutigkeit bei Ableitungen, die die anderen Verfahren meist nicht erreichen und in der Möglichkeit, dies auch noch zu automatisieren. Allerdings ist es sehr schwierig, heuristisches Wissen in diesen Prozeß einzubringen. Vielleicht ist dies der Grund, weshalb Logik bisher nicht zur Wissensrepräsentation in der Bildanalyse verwendet wurde.

Produktionen:

In Produktionensystemen ist das Wissen in einer Form niedergelegt, die direkt die Regeln zum Schließen enthält. Ein Produktionensystem besteht aus drei Teilen: einer Datenbasis, Produktionen und einem Interpreter. In der Datenbasis ist der derzeitige Zustand des Systems und die dort vorhandenen Daten abgespeichert. Auf diese Datenbasis greifen die Produktionen zu. Diese sind Regel der Form

IF Bedingung THEN Aktion.

Bei den Bedingungen kann es sich um einfache Aussagen oder auch sehr komplexe, strukturierte Regeln handeln (siehe z.B. /PUP 86/). Zu jedem Zeitpunkt können alle Produktionen, deren Bedingung zutrifft feuern, d.h. ihre Aktion zur Ausführung bringen. Da meist mehrere Produktionen gleichzeitig feuern könnten, muß der Interpreter entscheiden, welche der Produktionen tatsächlich zum Vollzug kommt. Produktionensysteme vollführen also folgenden Ausführungszyklus:

1. Bestimme (alle) Produktionen, die feuern können.
2. Wähle eine Produktion davon aus.
3. Führe die entsprechende Aktion aus.

Beendet wird der Lauf, wenn entweder keine Produktion mehr feuern kann oder ein gewünschter Zustand der Datenbasis erreicht ist. Produktionensysteme haben den Vorteil, daß sie sehr modular sind. Normalerweise beeinträchtigt das Einführen einer neuen Regel die restlichen Produktionen überhaupt nicht. Da auch die Struktur ziemlich genau festgelegt ist, sind Produktionensysteme einfach zu handhaben. Allerdings werden sie bei größeren Systemen ineffizient, da zu jedem Zeitpunkt alle Produktionen getestet werden müssen.

Frames:

1975 erstmals von Minsky /MIN 75/ vorgeschlagen, ist es mit Frames besonders einfach, generisches Wissen, also Objektklassen, zu modellieren. Frames sind festgelegte Rahmen, in denen einzelne Einträge, sog. "slots", vorhanden sind, die mit Wissen auf verschiedenen Abstraktionsstufen gefüllt sein können. Als Beispiel sei hier ein generischer DURCHSCHNITT-Frame und eine Instanz davon aufgeführt:

```
Generisch DURCHSCHNITT Frame
          Spezialisierung-von  :  VERKNÜPFUNG
          1. CSG-Baum          :  Name
          2. CSG-Baum          :  Name
          Voraussetzungen      :  REGULARISIERT

          DURCHSCHNITT1 Frame
          Spezialisierung-von  :  DURCHSCHNITT
          1. CSG-Baum          :  WÜRFEL 1
          2. CSG-Baum          :  WÜRFEL 2
          Voraussetzungen      :  REGULARISIERT
```

So arbeitet man mit einer Grundmenge an generischen Frames als Modell. Bei der Analyse werden dann für die gefundenen Objekte Frames erzeugt, die die Slots der generischen Frames erben, in denen aber die gefundenen Werte eingetragen werden.

Semantische Netze:

Semantische Netze, auch assoziative Netzwerke genannt, sind in Bildanalysesystemen weit verbreitet. Ein Netz besteht aus Knoten, in denen Objekte, Ereignisse oder ähnliches dargestellt sind und Kanten, die die Beziehungen der Knoten untereinander ausdrücken. Hierzu tragen die Kanten Bezeichnungen wie IS-A, PART-OF o.ä. Bild 4.1 zeigt ein semantische Netz mit einem Knoten Durchschnitt -*, der zwei Teile hat, die je ein CSG-Baum sind. Jeder der beiden CSG-Bäume hat einen Würfel als Spezialisierung, d.h. der Würfel ist ein Spezialfall des CSG-Baums. Der große Vorteil der semantischen Netze liegt in den ihnen innewohnenden Hierarchisierungen. Bei diesem Beispiel sind die beiden gängisten

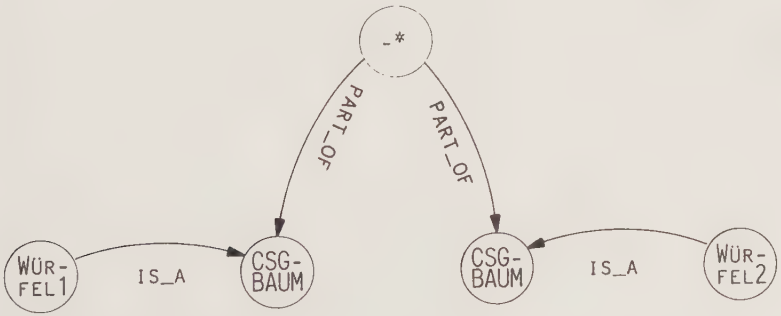


Bild 4.1: Semantisches Netz

Hierarchisierungen aufgeführt, über die PART-OF- oder über die IS-A-Kanten. Dies leitet die Suche im Netz während des Analyselaufs und vermeidet die vollständige Suche über eine komplette Datenbasis.

Die vier bis jetzt aufgeführten Repräsentationsformalismen stammen aus der künstlichen Intelligenz und werden dort eingehend studiert. Nun folgen noch drei Verfahren, die in der Bildanalyse Verwendung finden.

Formale Grammatiken

Kann man die Menge der Modelle zurückführen auf eine Menge primitiver Elemente, die nach speziellen Regeln zusammengefaßt werden, so kann man dies durch eine formale Grammatik beschreiben. Eine formale Grammatik G besteht aus einer Menge T_G terminaler Symbole. Dies sind die primitiven Elemente. Weiterhin gehört zu G eine Menge nichtterminaler Symbole N_G, die vor allem als Platzhalter fungieren, ein Startsymbol $S \in N_G$ und eine Menge R_G von Ableitungsregeln. Eine Grammatik G definiert nun eine formale Sprache L_G, die alle aus S durch beliebige Anwendung von Regeln ableitbaren Wörter $w \in T_G^*$ enthält. Die Syntax der CSG-Bäume könnte etwa mit folgender Grammatik geschrieben werden:

S → CSG-Baum
CSG-Baum → Primitives Element
CSG-Baum → CSG-Baum $\cup^*$ CSG-Baum
CSG-Baum → CSG-Baum $\cap^*$ CSG-Baum
CSG-Baum → CSG-Baum $-^*$ CSG-Baum
CSG-Baum → CSG-Baum M Bewegungs-Parameter

Je nach Aufbau der Regeln kann man Grammatiken unterscheiden, bei deren Sprachen es entscheidbar oder unentscheidbar ist, ob ein Wort $w \in T_G^*$ zu dieser Sprache gehört, oder nicht (siehe z.B. /SCN 75/). Formale Grammatiken haben breiten Eingang in die Mustererkennung und in die Bildanalyse gefunden, da es inzwischen gute Algorithmen (Parser) gibt, um ein aufgenommenes Muster (Bild) zu analysieren (siehe z.B. /FU 82/, /NIE 83a/).

Relationale Datenbank:

Eine relationale Datenbank (vgl. /DAT 81/, /WED 78/) bietet zur Abspeicherung der Daten die Relation als Datenstruktur. Dies ist eine Tabelle, die aus einer Anzahl von Spalten, sog. Attributen, besteht, deren Reihenfolge signifikant ist. Jedes Attribut hat einen Definitionsbereich, aus dem Werte in das Attribut eingetragen werden können. Jede Relation beinhaltet nun eine Menge von Zeilen, die Tupel heißen und je einem Eintrag entsprechen. In Bild 4.2 sind einige Relationen mit ihren Attributen angegeben. Zum Zugriff auf diese Relationen

kann man z.B. die sog. relationale Algebra als Abfragesprache verwenden. Diese umfaßt die bekannten Mengenoperationen Vereinigung, Durchschnitt, Differenz und Produkt sowie die Relationenoperationen Selektion (Auswahl von Tupel), Projektion (Auswahl von Attributen), Verbund (Zusammenschluß von zwei Relationen) und Division (Division zweier Relationen). Relationale Datenbanken wurden ursprünglich für die Verwaltung konzipiert, finden aber in neuerer Zeit immer mehr Eingang in Nicht-Standard-Anwendungen wie CAD /EBL 84/ und Bildanalyse /BEN 85/.

```
RELATION VERKNÜPFUNG
  CSG-BAUM      CSG-BAUM
RELATION BEWEGUNG
  CSG-BAUM      TRANSLATION  ROTATION
RELATION PRIMELEMENTE
  CSG-BAUM      PARAMETER    PARAMETER    PARAMETER
```

Bild 4.2: Relationen

Listen:
Die einfachste Möglichkeit, Modelle abzuspeichern, ist die Form der Liste. Dies ist vor allem dann sinnvoll, wenn die Modelle sehr wenig Struktur beinhalten und nicht zu groß sind. Sind die Objektmodelle aus einzelnen Kennzahlen aufgebaut, so können diese so aufgelistet werden, daß durch die Anordnung die Interpretation der Daten unterstützt wird. Diese Methode verwenden z.B. Yachida und Tsuji /YAC 77/ in ihrem System zur Erkennung von Werkstücken. Der Vorteil der Listen ist die schnelle Art, in der auf einzelne Elemente zugegriffen werden kann.

4.3 Einsatz und Vergleich verschiedener Ansätze zur Wissensrepräsentation

Tabelle 3.2 und Tabelle 3.3 in Kapitel 3 führen auf, welche Wissensrepräsentationsformalismen in einigen Bildanalysesystemen eingesetzt werden (jeweils in der Spalte Darstellung). Dabei wird, wie schon in Kapitel 3, zwischen Objektmodell und Gesamtmodell unterschieden. Am häufigsten werden dort Listen oder

Tabellen verwendet, in denen einzelne Merkmale abgelegt sind. Diese Form ist einfach zu implementieren und trotzdem schnell, wenn die Menge der Merkmale nicht zu groß ist. Für ein allgemeines Bildanalysesystem sind Listen und Tabellen zur Abspeicherung des Modells aber so unstrukturiert, daß ein schneller Zugriff schwierig ist. Auch semantische Netze sind ziemlich verbreitet, da sie der "natürlichen" Strukturierung des Wissens entgegenkommen und den Analyselauf unterstützen. Einen Modellgraphen, in dem Objektmodelle und Objektklassen codiert sind, baut Brooks /BRO 81/ auf. Daraus leitet er einen Vorhersagegraphen ab, der die Invarianten des Modells beinhaltet. Die Interpretation läuft dann über einen Teilgraphvergleich zwischen Vorhersagegraph und Beschreibungsgraph, der aus dem Bild mittels Bildverarbeitungsalgorithmen gewonnen wird. Auch Rubin /RUB 78/ baut einen Graphen auf, in dem alle möglichen Bilder in einem Vorlauf hineincompiliert werden. Damit hat er ein HARPY-ähnliches Netz /LOW 80/ erzeugt, in dem beim Analyselauf nur noch ein Pfad vom Anfangsknoten zu einem Endknoten gesucht werden muß. Dieser Ansatz beschränkt allerdings Größe und Variabilität des Modells durch die maximal noch berechenbare Netzgröße. Mehrfach kommen Produktionen oder Prozeduren zum Einsatz. Dies ist bisher bei größeren Systemen noch kaum erforscht. Durch den derzeitigen Trend zu Expertensystemen, die mit ähnlichen Verfahren arbeiten, entstehen aber auch in der Bildanalyse vermehrt Systeme, die sich auf Regeln und Produktionen zur Wissensrepräsentation stützen /NEE 85/. Für das beschriebene Bildanalysemodellierungssystem stellte sich die Frage zwischen semantischem Netz und relationaler Datenbank. Für das semantische Netz sprach die bessere und klarere Gliederung des Wissens. Man hat dort eine natürliche Möglichkeit, Hierarchien aufzubauen. Daneben wird auch die Analyse unterstüzt, da sie entlang den im Netz vorhanden Kanten vorgehen kann. Die relationale Datenbank zeichnet sich durch den schnellen und wahlfreien Zugriff auf alle Relationen und deren Attribute aus. Man kann direkt auf Teile von Objekten und auch auf Oberflächenattribute zugreifen, ohne das Objektmodell kennen zu müssen. Auch Änderung und Ergänzungen sind ohne Auswirkung auf die restliche Datenbank möglich. Dies gab zusammen mit fachfremden Argumenten letzlich den Ausschlag für die Verwendung einer relationalen Datenbank zur Wissensrepräsentation.

5. Ein Bildanalysemodellierungssystem

Die Strukturen und Algorithmen, die benötigt werden, um ein Modell in einem Bildanalysesystem (vgl. Bild 2.2) aufzubauen, seien Bildanalysemodellierungssystem (BMS) genannt. Wie schon in Abschnitt 2.1 ausgeführt, sind zum Aufbau eines BMS zwei Schritte nötig. Bild 5.1 verdeutlicht dies nochmals und gibt die dazugehörigen "Daten" an. Ausgangspunkt sind Objekte aus einem Problembereich und deren Beziehungen zueinander. Sowohl die Objekte als auch die zugehörigen Beziehungen müssen in einem ersten Schritt, den wir Modellbildung nennen, in mathematisch beschreibbare Modelle umgesetzt werden. Im zweiten Schritt, dem Repräsentationsschritt, müssen diese mathematischen Modelle geeignet in einem Computer gespeichert werden. Ein BMS muß nun die Strukturen und Algorithmen bereitstellen, die für diese beiden Schritte benötigt werden. Dies ist zum einen ein Verfahren, das es ermöglicht, Objekte der realen Welt und ihre Beziehungen quantitativ zu erfassen. Solche Verfahren wurden bereits in Kapitel 3 unter dem Begriff geometrische Modellierer allgemein erläutert. Zum andern umfaßt ein BMS einen Formalismus zur Wissensrepräsentation, eben zur Darstellung der mathematisch beschreibbaren Modelle auf dem Rechner. Aus diesen beiden Komponenten setzt sich ein BMS zusammen. In Kapitel 5 soll nun ein spezielles BMS ausführlich vorgestellt werden.

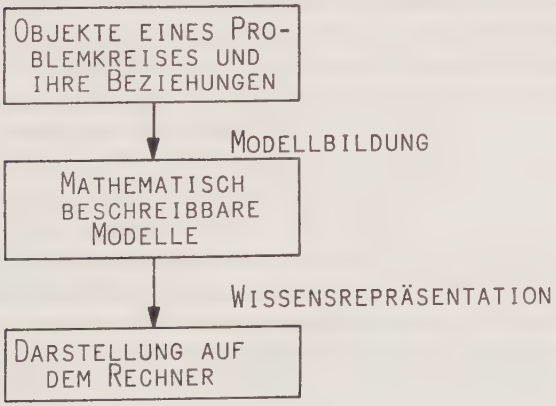


Bild 5.1: Aufbau eines Bildanalysemodellierungssystems

5.1 Anforderungen an ein Bildanalysemodellierungssystem

Im Abschnitt 3.2 wurden verschiedene Kriterien diskutiert, die bei Auswahl und Bewertung eines BMS verwendet werden können. Die Kriterien, denen das hier vorgestellte BMS genügen soll, kann man grob in drei Gruppen gliedern. Sie beziehen sich entweder auf die geometrische Modellierung oder auf Bildanalyseaspekte oder sie sollen die Performanz des Systems verbessern.

Zu der ersten Gruppe zählt die Forderung nach Standpunktunabhängigkeit. Damit soll erreicht werden, daß nicht für verschiedene Ansichten eines Objektes verschiedene Objektmodelle im Rechner angelegt werden müssen. Auf der Basis von zweidimensionaler Modellierung versucht das Rubin /RUB 78/ durch das Zulassen von Variationen in den örtlichen Beziehungen und Verdeckungen der modellierten Häuser (siehe Abschnitt 3.4). Universell ist diese Forderung aber nur durch ein dreidimensionales Modellierungssystem zu erfüllen. Weiterhin erhöht ein hierarchischer Aufbau von Objektmodellen die Einfachheit des geometrischen Modellierers. Diese wiederum sollte in einem BMS als Entwurfskriterium berücksichtigt werden, um einerseits den Umgang mit dem System nicht zu schwierig zu gestalten und andererseits die Entwicklung einer zusätzlichen Komponente, die die Objektmodelle lernen soll, zu erleichtern.

Weitere drei Forderungen zielen vor allem auf Bildanalyseaspekte. Für die Bildanalyse ist es notwendig, daß die aus der Bildvorverarbeitung gelieferten Daten mit den vom Modell bereitgestellten Daten für Interpretationszwecke vergleichbar sind. Aus heutiger Sicht benötigt man deshalb eine Möglichkeit, zweidimensionale Daten zu verarbeiten oder aus dem Modell abzuleiten. In manchen Problembereichen, wie z.B. natürlichen Szenen, existiert eine Vielfalt ähnlicher Objekte. Darüberhinaus können auch einzelne Objekte über einen weiten Bereich variieren. Man denke dabei nur an verschiedene vierbeinige Stühle. Hierfür eignet sich die Definition von Objektklassen, bei denen dann nur ein Objektmodell für eine ganze Menge ähnlicher Objekte gespeichert werden muß. Wesentliche Merkmale bei der Interpretation von Bildern sind "visuelle" Attribute der Objekte, also die Eigenschaften eines Objekts, die die Kamera aufnimmt. Deshalb muß es möglich sein, solche Attribute für die Oberflächen eines Objektes zu definieren. Dabei sollte eine Erweiterung der Attribute auf neue Merkmale ebenso möglich sein, wie ein schneller Zugriff auf bestimmte Werte eines Attributs.

Die letzte Gruppe der Kriterien, die zu beachten sind, sollen die Performanz

des BMS verbessern. Hierzu kann eine Hierarchisierung der Szene beitragen, da daraus Hinweise für eine wissensgesteuerte Analyse abgeleitet werden können. Ein Problem bei existierenden Bildanalysesystemen stellt oft die Tatsache dar, daß eine Erweiterung des Modells aus prinzipiellen Gründen nicht durchführbar ist. Eine solche Einschränkung sollte auf ein BMS nicht zutreffen. Stattdessen sollte eine Begrenzung der Anzahl der Objektmodelle nur aus Gründen des Speicherplatzes oder des Rechenzeitverbrauches bei der Analyse notwendig sein. Zuletzt sei noch darauf verwiesen, daß eine Änderung des Problembereichs oft auch eine Änderung der Strukturen des BMS bedingt. Dies wird sich z.B. in den zwei Anwendungsfällen in Kapitel 8 zeigen. Je weniger aufwendig deshalb eine Änderung oder Erweiterung des BMS ist, um so einfacher läßt sich dann ein Übergang zwischen verschiedenen Problembereichen bewerkstelligen.

5.2 3D-Modellierung mit verallgemeinerten Zylindern

Nach der Beschreibung von Anforderungen, die ein BMS erfüllen soll, wird in diesem und den folgenden Abschnitten ein BMS vorgestellt, das zumindest den meisten dieser Bedingungen genügt. Zuerst wird die geometrische Modellierung von Objekten eingeführt. Hierbei bauen die Objektmodelle auf einfachen geometrischen Körpern, die im weiteren Primelemente genannt werden, auf. Als Primelemente wurden verallgemeinerte Zylinder (VZ) ausgewählt. Die allgemeine Definition eines VZ findet sich in Kapitel 3. Wie bereits dort schon vermerkt, ist diese Definition für die Verwendung in einem BMS ungeeignet, da sie für die Verarbeitung zu aufwendig ist, selbst wenn man eine diskrete Version davon nimmt. Deshalb werden in Bildanalysesystemen spezialisierte VZ verwendet. Einige dieser Spezialisierungen wurden in Kapitel 3 angegeben. Entsprechend den oben formulierten Forderungen wird nun Objektmodellierung auf der Grundlage einer veränderten Definition von VZ vorgestellt.

5.2.1 Definition der Primelemente

Ein VZ wird beschrieben durch Grundfläche, Deckfläche, Höhe und lokales Koordinatensystem (KOS). Wie diese Elemente einen VZ definieren, ist aus Bild 5.2 zu ersehen. Dabei steht die Höhe für die Achse aus der allgemeinen Definition für einen VZ. Entsprechend dieser Angabe kann die Achse also nur gerade sein und keine beliebige Kurve im Raum darstellen. Von der Form her unterschiedliche Primelemente lassen sich durch unterschiedliche Grund- und Deckflächen erreichen. Bild 5.3 zeigt drei verschieden VZ, die auf Viereck, Kreis und Dreieck als Grundfläche aufbauen. Für die Definition eines Vierecks

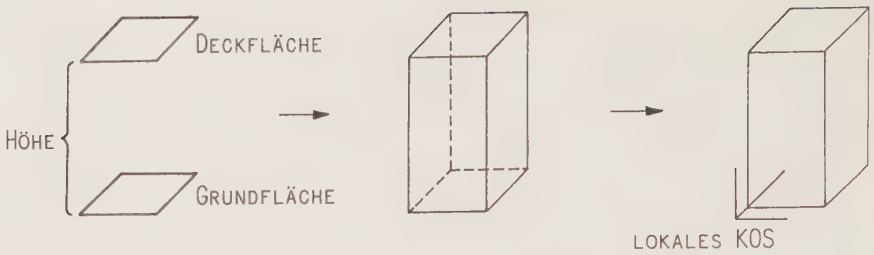


Bild 5.2: Definition eines verallgemeinerten Zylinders

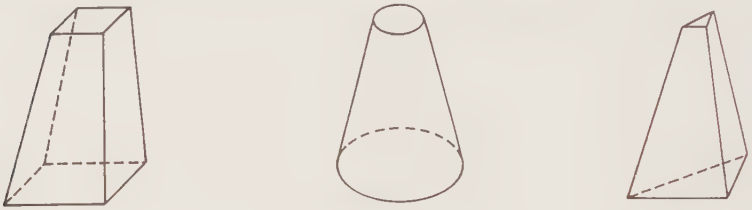


Bild 5.3: Verschiedene verallgemeinerte Zylinder

oder eines Dreiecks als Grundfläche sind vier bzw. drei Eckpunkte nötig. Diese werden in lokalen Koordinaten angegeben. Allein der Radius genügt zur Definition des Kreises als Grundfläche, da implizit angenommen wird, daß der Ursprung des lokalen KOS im Mittelpunkt des Kreises liegt.

Zusätzlich zu dieser Definition, teilweise auch direkt daraus abgeleitet, ergeben sich einige Einschränkungen.

- Grundfläche und Deckfläche müssen die gleiche Form haben, da sonst zu komplizierte Primelemente entstehen.
- Grundfläche und Deckfläche müssen parallel liegen, da nur der Abstand zwischen beiden als Höhe in die Definition eingeht.
- Die Oberflächen dürfen nur planar oder zylindrisch sein. Dies schränkt zwar den Definitionsbereich ein, vereinfacht aber die Berechnung und erscheint für die Bildanalyse eine zulässige Beschränkung.
- VZ mit Kreis als Grund- und Deckfläche können nicht verschoben sein aufgrund der Gleichsetzung von Koordinatenursprung und Kreismittelpunkt. Bei VZ mit einem Viereck als Grund- und Deckfläche ist dies möglich (siehe Bild 5.4).

Für VZ mit Kreis als Grund- und Deckfläche existiert (aus traditionellen Gründen) noch eine alternative Definitionsmöglichkeit. Wie in Bild 5.5 gezeigt

wird, werden hierzu Grundfläche, Höhe, eine Veränderungsregel ("sweeping rule") und lokales KOS verwendet. Die Veränderungsregel gibt dabei den Radius des Kreises als Funktion von der Höhe des Zylinders an. Wenn also wie in Bild 5.5 die Veränderungsregel lautet

$$y = ax + b$$

und die Höhe ist x cm, dann hat die Grundfläche einen Radius von b cm, die Deckfläche einen Radius von (a*Höhe + b) cm und dazwischen liegt ein kontinuierlicher Übergang. Die Veränderungsregel muß eine konstante oder eine lineare Funktion sein, so daß die in Bild 5.6 gezeigten Primelemente nicht zulässig sind.

5.2.2 Definition von Objekten

Aus den so definierten Primelementen werden dann Objektteile und Objekte zusammengesetzt. Dabei können Objekte aus einer beliebigen Mischung von Objektteilen und Primelementen bestehen. Zum Aufbau der Objekte wird der Vereinigungsoperator verwendet. Dieser Vereinigungsoperator wird durch die

Bild 5.4: Verschobene verallgemeinerte Zylinder

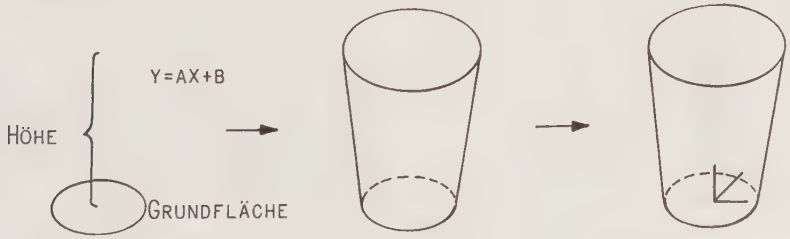


Bild 5.5: Alternative Definition für zylindrische VZ

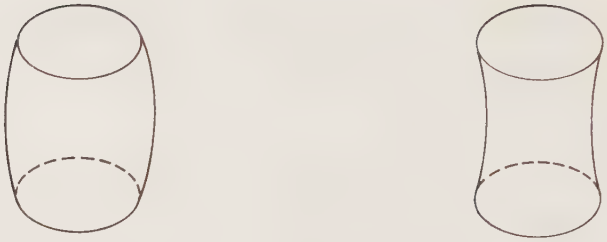


Bild 5.6: Nicht zulässige verallgemeinerte Zylinder

Transformation der lokalen KOS in ein Objektteil-KOS bzw. in ein Objekt-KOS ausgedrückt. Mit sechs Parametern, drei Translations- und drei Rotationsparametern, läßt sich diese Transformation beschreiben. Bild 5.7 zeigt zwei einfache Objekte in einer Drahtgitterdarstellung. Der Stuhl auf der linken Seite wird aus sechs Primelementen gebildet. Dabei sind vier dieser Primelemente (Stuhlbeine) identisch und unterscheiden sich nur durch ihre örtlichen Beziehungen innerhalb des Objekt-KOS. Damit genügt es, dieses Primelememt nur einmal zu definieren, sozusagen einen Prototyp davon zu haben, und dann lediglich im Modell zu vermerken, daß vier solche Primelemente zum Objekt Stuhl gehören. Zu jedem Stuhlbein muß noch die Transformation gespeichert werden, die die örtliche Lage dieses speziellen Objektteils im Bezug zum Objekt angibt. Ähnliches gilt für das Objektmodell des dreibeinigen Hockers, der auf der rechten Seite im Bild 5.7 abgebildet ist. Die Hockerbeine sind übrigens Beispiele für verschobene VZ, wie sie im Abschnitt 5.2.1 beschrieben wurden.

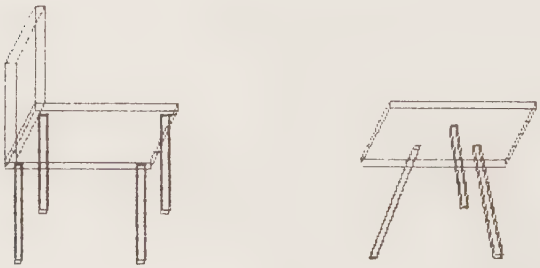


Bild 5.7: Zwei Objektmodelle

Die Gültigkeit der Objekte ist bei dieser Art der Definition von den Primelementen und dem Verknüpfungsoperator abhängig, wie bereits im Abschnit 3.2 ausgeführt wurde. Dies ist neben der Vereinfachung der Verarbeitung einer der Hauptgründe, weshalb hier als Verknüpfungsoperation nur die Vereinigung zugelassen wurde. Damit entfällt beim Aufbau der Objekte die Regularisierung, die bei der Anwendung von Durchschnitt oder Differenz als Verknüpfungsoperation erforderlich ist (siehe Kapitel 3). Arbeitet man mit einfachen Primelementen, so kann deshalb ein Test der Gültigkeit entfallen. Allerdings ist nicht jedes geometrisch plausible Objekt auch ein im Kontext des Problembereichs sinnvolles Objekt. Deshalb sind zusätzliche Bedingungen für solche Problemfälle nötig, die im fünften Abschnitt dieses Kapitels eingeführt werden.

Der Hauptkritikpunkt an der Modellbildung mit VZ ist die Tatsache, daß der Definitionsbereich der VZ zu gering ist. Dies führen z.B. Besl und Jain /BES 85/ oder Ballard und Brown /BAL 82/ dagegen an. Andererseits ist der Ansatz einfach und klar zu gliedern, und es kommt gerade in der Bildanalyse oft nicht auf jedes kleinste Detail an, was eine Modellbildung mit VZ dann möglich macht. So finden VZ bei vielen Forschern, z.B. /AGI 73/, /NIS 81/, /SHA 80/, in Bildanalysesystemen Verwendung. Um den Definitionsbereich des hier vorgestellten BMS zu erweitern, besteht die Möglichkeit, bei Bedarf eigene Primelemente zu definieren. Ein solches neues Primelement könnte z.B. ein Quader mit einem Loch in der Mitte sein. Wie in Bild 5.8 gezeigt, benötigt man hierzu die entsprechende Definition von Grund- und Deckfläche, die Angabe der Höhe und das lokale KOS. Grund- und Deckfläche werden hierbei durch die vier Eckpunkte in lokalen Koordinaten, den Ursprung in lokalen Koordinaten und dem Radius des Kreises gegeben. Zusätzlich muß allerdings sichergestellt sein, daß das Primelement nicht wie in Bild 5.9 durch falsche Parameterwerte in mehrere Teile zerfallen

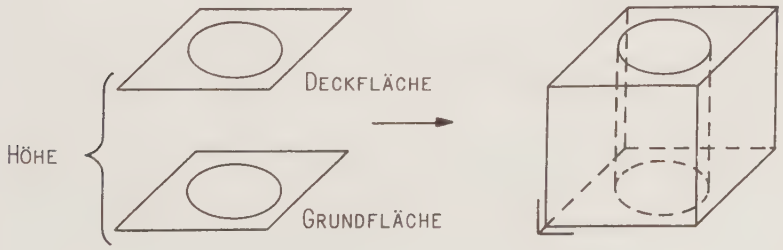


Bild 5.8: Ein weiteres Primelement

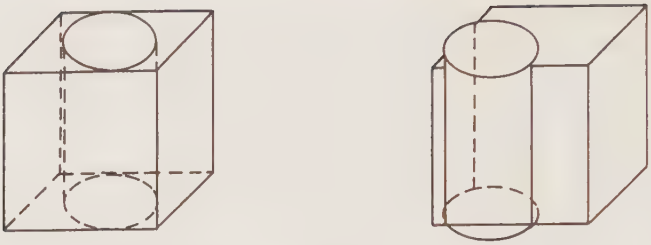


Bild 5.9: Nicht zulässige Primelemente

kann. Mit dieser zusätzlichen Möglichkeit erweitert sich der Definitionsbereich des BMS. Wie davon Gebrauch gemacht werden kann, wird in Kapitel 8 bei den Anwendungsbeispielen noch erläutert werden.

Weitere Schwierigkeiten können dadurch entstehen, daß keine Eineindeutigkeit gegeben ist, da Symmetrie weder in den Primelementen noch bei den Objekten berücksichtigt ist. Dies führt dazu, daß für ein Objekt mehrere Definitionen möglich sind. Überlegungen zu dem Problem der Modellierung symmetrischer Objekte werden z.B. in /RAD 85/ angestellt. Allerdings sind diese nicht direkt übertragbar. Deshalb wird eine Betrachtung der Symmetrie der Objekte bei der Modellbildung ausgeklammert. Dieses Problem muß dann bei der Analyse durch geeignete Einbeziehung in den Kontrollalgorithmus gelöst werden. Zuletzt sei noch darauf hingewiesen, daß bei der Zerlegung der Objekte in Primelemente Kanten im Modell auftreten können, die im Bild nicht vorhanden sind. Auch dieses Problem muß während der Analyse gelöst werden. Wie eine Lösung bereits bei der Modellbildung aussehen könnte, wird im nächsten Kapitel bei einem alternativen Ansatz für ein BMS gezeigt.

5.2.3 Definition einer Szene

Nachdem im vorherigen Abschnitt die Erstellung von Objektmodellen aus VZ diskutiert wurde, wird nun die Zusammenfassung mehrerer Objektmodelle besprochen. Hierfür stehen sogenannte Objektgruppen zur Verfügung. Eine Objektgruppe besteht aus zwei oder mehreren Objekten, die meist (oder immer) gemeinsam in Bildern des Problemkreises auftreten. Zur Beschreibung einer Objektgruppe sind ein ausgezeichnetes Objekt, das das Objektgruppen-KOS trägt ein oder mehrere Objekte und deren örtliche Beziehungen notwendig. Das ausgezeichnete Objekt wird auch Hauptobjekt einer Objektgruppe genannt. Die örtli-

chen Beziehungen der Objekte werden analog zum Aufbau eines einzelnen Objekts wieder durch die Transformationen der Objekt-KOS in das Objektgruppen-KOS bestimmt. Ein einfaches Beispiel für eine Objektgruppe sind ein Schreibtisch und ein Schreibtischstuhl in einer Büroszene (siehe Bild 5.10). Mit der Konstruktion von Objektgruppen ist eine Hierarchisierung der Szene erreicht. Außerdem können Objektgruppen dazu verwendet werden, bei der Analyse Hypothesen mithilfe des Modells ("top-down") zu erzeugen, wenn man erst einmal ein Objekt einer Objektgruppe erkannt hat. Deshalb sollten Objektgruppen aus diesem Grund auch nur dort modelliert werden, wo eine hohe Wahrscheinlichkeit für ein gemeinsames Auftreten der beteiligten Objekte besteht.

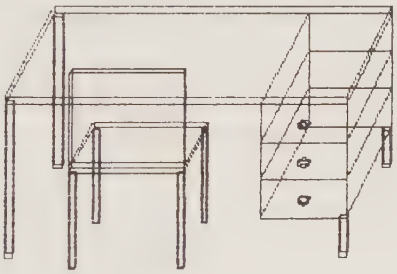


Bild 5.10: Eine Objektgruppe

Das gesamte Modellwissen über einen Ausschnitt des Problembereichs wird in einer Szene zusammengefaßt. Eine Szene enthält z.B. das Modell eines Büros. In einer Szene ist also mehr Information, als für die Analyse eines Bildes benötigt wird. Sie besteht aus Objekten, Objektgruppen und deren örtlichen Beziehungen untereinander. Analog der Objektgruppe gibt es ein Hauptobjekt oder eine Hauptobjektgruppe, die das Szenen- KOS trägt. Die örtlichen Beziehungen werden dann wieder mittels entsprechenden Transformationen angegeben. Zusätzlich sollte es möglich sein, auch die Kamera in eine Szene und damit in die Szenenkoordinaten einzubeziehen. Ein Beispiel für eine Szene wird in Kapitel 8 gezeigt.

Ein Problem bei der 3D-Modellbildung stellen Objekte dar, die nicht sinnvoll in dreidimensionale Körper zu fassen sind. Dies können der Himmel in einer Vorstadtszene oder auch der Hintergrund (Fließband) in einer Werkstückszene sein. Um hier Abhilfe zu schaffen, können in einer Szene zusätzlich Regionen definiert werden. Für eine solche Region ist es meist nicht sinnvoll, eine bestimmte Form zu modellieren, denn diese Form ist wesentlich von den übrigen

Objekten im Bild abhängig. Vielmehr wird eine Interpretation über Intensität, Farbe o.ä. erfolgen müssen und kann eventuell durch die örtliche Stellung im Bild unterstützt werden. Entsprechend können im vorgestellten BMS für Regionen Attribute definiert werden analog zur Beschreibung im nächsten Abschnitt. Die im Kapitel 8 vorgestellte natürliche Szene beinhaltet auch einige Regionen als Beispiel.

Damit haben wir den geometrischen Teil der Definition von Szenen, Objektgruppen und Objekten vorgestellt. Aufbauend auf VZ als Primelemente können Objekte und ihre örtlichen Nachbarschaften modelliert werden. Über mehrere Stufen ist dabei eine Hierarchie gegeben. Die geometrische Modellbildung ist aber nur ein Teil dessen, was man in einem BMS benötigt. Deshalb werden jetzt weitere Teile im BMS erläutert.

5.3 Oberflächenattribute

Ziel für die Erstellung eines BMS bleibt die Analyse von Bildern. Diese Bilder können je nach Art des Aufnahmegeräts unterschiedlich sein. Bei einer Fernsehkamera - und darauf wollen wir uns hier beschränken - besteht ein bereits digitalisiertes Bild aus Intensitätswerten in einem oder mehreren Kanälen. Verschiedene Bildverarbeitungsmethoden gewinnen daraus abgeleitete Daten, die dann für Regionen im Bild gelten, wobei diese Regionen gerade dadurch definiert sind, daß innerhalb der Region diese abgeleiteten Werte homogen sind, während am Rand gewisse Sprünge auftreten. Solche "Homogenitätskriterien" können die Intensität oder die Farbe sein, aber auch komplexere Werte wie Sättigung oder Textur. Um nun eine Analyse des Bildes anhand des Modells zu ermöglichen, muß das Modell diese Informationen ebenfalls bereitstellen. Dazu können im Modell die Oberflächen der VZ mit diesen Informationen versehen werden, die wir dann Oberflächenattribute nennen.

Zur Kennzeichnung werden die Oberflächen der VZ durchnumeriert (siehe Bild 5.11). Die Grundfläche bekommt die Nummer 1, und die Deckfläche wird mit der höchsten Ziffer bezeichnet, die von VZ zu VZ unterschiedlich ist. Die Seitenflächen werden beginnend von der lokalen x-z-Ebene im Gegenzeigersinn durchgezählt. Somit hat ein Quader sechs Oberflächen, ein Pyramidenstumpf fünf und ein Zylinder drei.

Im Modell können nun den Oberflächen der VZ Attribute zugewiesen werden. Dabei sind vorerst nur Intensität, Farbe und Textur vorgesehen. Jede Oberfläche kann

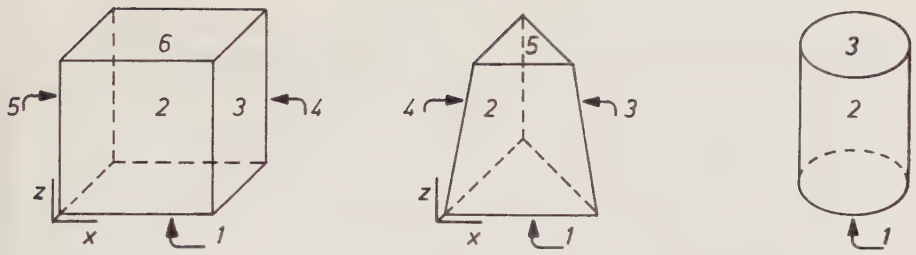


Bild 5.11: Numerierung der Oberflächen von VZ

mit allen drei oder auch mit weniger Attributen modelliert werden. Weiterhin besteht die Möglichkeit, einer Oberfläche verschiedene Werte eines Attributs zu geben. Diese verschiedenen Werte sind dann mit logischem ODER verknüpft, d.h. es kann einer von beiden oder auch alle beiden Werte zutreffen. Man kann sich nun noch viele weitere Oberflächenattribute denken, die von Bildverarbeitungsmethoden errechnet werden. Deshalb ist die Liste der bisher realisierten Oberflächenattribute nicht abgeschlossen, sondern man kann jederzeit ein neues Oberflächenattribut in das BMS einbringen. Wie dies ohne große Änderungen an dem bestehenden BMS möglich ist, werden wir im letzten Abschnitt dieses Kapitels sehen.

5.4 Bildung von Objektklassen

Eine wesentliche Forderung an ein BMS ist die Möglichkeit, Objektklassen zu bilden. Damit erhält man Gelegenheit, Objekte, die bei der Modellerstellung nicht bekannt sind, doch zu erkennen, sofern sie gewisse Ähnlchkeiten zu einem bekannten Objekt aufweisen. Diese Eigenschaft ist vor allem bei der Analyse von natürlichen Szenen wichtig. Es bietet sich an, die Vorgehensweise ähnlich den vier Schritten zu wählen, die Zadeh /ZAD 85/ in der "test score semantics" formuliert hat, um die Bedeutung einer Behauptung darzustellen. Es muß zuerst geklärt werden, welche Variablen in die Objektklassenbildung eingehen sollen, und welchen Eigenschaften oder Parameter dieser Variablen zur Klassenbildung beitragen sollen. Für jede Eigenschaft bzw. jeden Parameter muß eine Testfunktion gefunden werden, die einen Wert für das Zutreffen dieser Eigenschaft berechnen läßt. Zuletzt will man dann diese Einzelwerte noch geeignet zu einem Gesamtwert zusammenfassen, der in unserem Fall dann die Sicherheit angeben soll, daß eine in einem Bild gefundene Struktur ein Objekt aus einer Objektklasse des Modells darstellt. Um zu begründen, welche Teile des Modells zur

Objektklassenbildung herangezogen werden und mittels welcher Parameter dieser Teile die Objektklassenbildung durchgeführt werden soll, betrachten wir die Arten der Klassenbildung, die wir zulassen.

5.4.1 Arten der Klassenbildung

Bildanalyse wäre wesentlich einfacher, wenn jedes aufgenommene Objekt immer exakt dem zugehörigen Objektmodell entspräche. Aufgrund von drei Hauptfehlerquellen ist dies aber nicht so. Die erste Fehlerquelle betrifft das Bild. Sowohl bei der Aufnahme, als auch bei der Verarbeitung können Fehler entstehen, die eine Abweichung vom Objektmodell bewirken. Solche Fehler, die bei der Aufnahme entstehen, sind oft statistischer Natur, und wir wollen diese hier nicht weiter betrachten, zumal es zu ihrer Beseitigung oft bessere Algorithmen gibt als ein modellgesteuertes Verfahren. Weiterhin kann das Objekt Toleranzen bei der Herstellung, durch Beschädigung erlittene Änderungen oder ähnliches aufweisen. Auf diese Art von Fehlern werden wir im Kapitel 8 nochmals eingehen. Als drittes kann man sich vorstellen, Objektvariationen, die auf verschiedene Objekte zurückgehen, in eine Objektklasse zu modellieren, wie eingangs schon erwähnt. Im weiteren werden die dabei möglichen Objektvariationen untersucht.

Beim geometrischen Teil der Modellbildung werden vier Variationsarten berücksichtigt. Bild 5.12 zeigt dazu mögliche Ausprägungen eines Objektmodells, in denen die verschiedenen Variationsarten realisiert sind. Zuerst wird man die Variation der Ausmaße zulassen. Eine Verlängerung des Hauses soll nicht die Generierung eines neuen Objektmodells bedingen. Ebenso möchte man kleine Formänderungen innerhalb einer Objektklasse tolerieren. Hierbei gilt allerdings im besonderen, was beim Aufbau eines Modells mit Objektklassen beachtet werden muß. Die Grenze dessen, was noch innerhalb einer Objektklasse zusammengefaßt werden kann, ist von vielen Faktoren, nicht zuletzt vom Anwendungsfall abhängig. Oft enthalten Objekte gleichartige Teile, deren genaue Anzahl von Objekt zu Objekt schwanken kann, wie z.B. die Anzahl der Fenster in Bild 5.12 d. Wenn diese Anzahl für die Interpretation nicht von vorrangiger Bedeutung ist, muß diese innerhalb einer Objektklasse variieren können. Zuletzt sollte auch eine Variation der örtlichen Beziehungen von Objektteilen innerhalb eines Objekts (Bild 5.12 e) oder Objekte einer Objektgruppe möglich sein.

Neben diesen Variationen der Geometrie von Objektmodellen soll auch eine Variation der im letzten Abschnitt eingeführten Oberflächenattribute erlaubt sein. Dabei handelt es sich vor allem um Variationen der Attribute Intensität

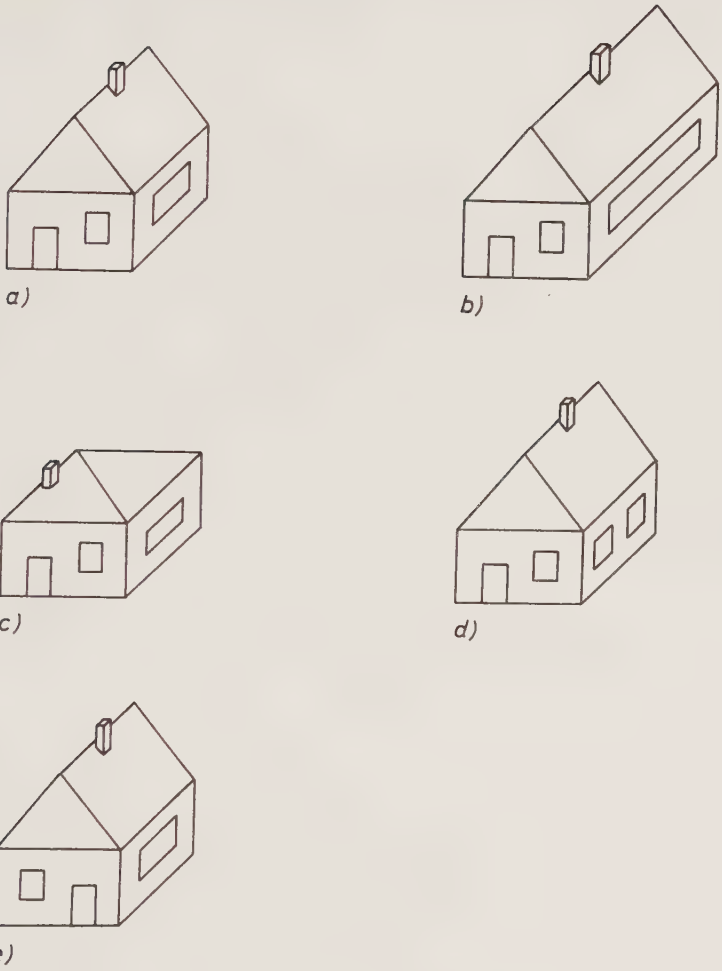


Bild 5.12: Objektvariationen

a) Originalobjekt
b) Variation der Ausmaße
c) Variation der Form
d) Variation der Anzahl der Objektteile
e) Variation der örtlichen Beziehungen

oder Farbe, die bei natürlichen Szenen ziemlich unterschiedlich sein können. Sollten weitere Oberflächenattribute in das Modell aufgenommen werden, so wird man auch dabei überlegen müssen, ob es sinnvoll ist, dort eine Variation der Werte zuzulassen.

5.4.2 Fuzzy-Funktionen

Bei der Bildanalyse geht es, wie schon erwähnt, darum, einer im Bild gefundenen Struktur mit einer gewissen Sicherheit ein Teil aus dem Modell zuordnen zu können. So stellt z.B. ein im Bild gefundener Kreis mit Radius 5 cm mit Sicherheit 0.8 die Grundfläche eines bestimmten VZ im Modell dar. Um dies bewerten zu können, verwenden wir im Modell Kompatibilitätsfunktionen auf der Basis von Fuzzy-Logik. Diese Kompatibilitäsfunktionen nennen wir Fuzzy-Funktionen. Eine andere Möglichkeit wäre, Kompatibilitätsfunktionen auf der Basis von Wahrscheinlichkeiten zu erstellen. Da es aber in manchen Problembereichen, wie z.B. natürlichen Szenen, schwierig ist, geeignete Wahrscheinlichkeitsfunktionen für die einzelnen Parameter zu schätzen, werden hier Fuzzy-Funktionen bevorzugt. Auch bei diesen kann es zwar manchmal schwierig sein, gesicherte Funktionen zu erhalten, aber es ist einfacher, auch mit mehr informell gefundenen Fuzzy-Funktionen zu arbeiten.

Eine formale Einführung in Fuzzy-Logik findet sich z.B. in /ZAD 75, ZAD 79, BEZ 81/ oder in gedrängter Form in /SAG 85/. Wir geben hier eine mehr informelle Begründung für die Art und die Verwendung von Fuzzy-Funktionen zur Objektklassenbildung.

Es sei eine Menge U gegeben. Zur Vereinfachung betrachten wir vorerst eine endliche Menge $U = \{u_1,\ldots,u_n\}$. Eine endliche Fuzzy-Teilmenge A von U lautet dann $\{\mu_1 u_1,\ldots,\mu_n u_n\}$. Dabei gilt $\mu_i \in [0,1]$.
Die μ_i drücken also den Grad der Zugehörigkeit eines Elements u_i zu der Fuzzy-Teilmenge A aus. Bei der herkömmlichen Mengenlehre kann dieser Grad nur gleich 0 oder gleich 1 sein. Dort kann ein Element u_i zu einer Teilmenge gehören oder eben nicht. Im Gegensatz dazu ist es bei Fuzzy-Teilmengen möglich, daß ein Element u_i nur mit einer bestimmten Sicherheit in einer Fuzzy-Teilmenge ist. Zum Beispiel ist U ={a,b,c,d,e } eine Menge und A = { a, 0.5b, 0.5c, 0d, 0.7e} eine Fuzzy-Teilmenge davon. Vereinbarungsgemäß läßt man die Elemente, für die $\mu_i = 0$ gilt, weg, so daß A dann lautet A = { a, 0.5b, 0.5c, 0.7d}. Faßt man die μ_i zu einer Funktion μ_A zusammen, so spricht man von einer Kompatibilitätsfunktion für die Fuzzy-Teilmenge A, die wir, wie schon erwähnt, Fuzzy-Funktion

nennen. Für solche Fuzzy-Funktionen gibt es einige gängige Parametrisierungen, die auf kontinuierlichen Definitionsmengen basieren. In Bild 5.13 ist eine Fuzzy-Funktion dargestellt, die durch die drei Parameter α, β, γ und folgende Gleichungen beschrieben wird.

$$S(u;\alpha,\beta,\gamma) = \begin{cases} 0 & u \leq \alpha \\ 2 \cdot \left(\frac{u-\alpha}{\gamma-\alpha}\right)^2 & \alpha \leq u \leq \beta \\ 1-2 \cdot \left(\frac{u-\gamma}{\gamma-\alpha}\right)^2 & \beta \leq u \leq \gamma \\ 1 & u \geq \gamma \end{cases} \quad \text{für}$$

Dabei ist β, $\beta = \frac{\alpha+\gamma}{2}$, der Umkehrpunkt.

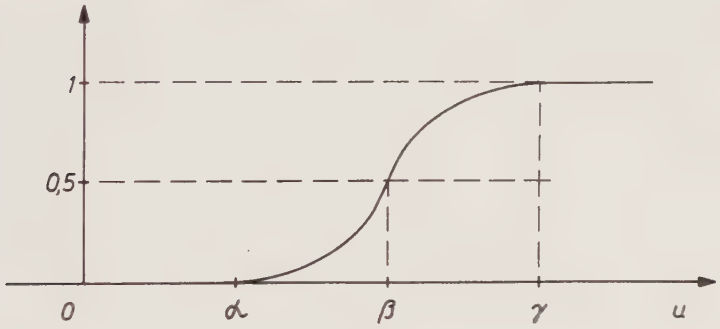


Bild 5.13: die Fuzzy-Funktion $S(u;\alpha,\beta,\gamma)$

Abhängig von den vier Parametern A,B,C,D ist die trapezförmige Fuzzy-Funktion F(u; A,B,C,D), die wir verwenden (siehe Bild 5.14).

$$F(u;A,B,C,D) = \begin{cases} 0 & u \leq A \\ \frac{u-A}{B-A} & A \leq u \leq B \\ 1 & B \leq u \leq C \\ 1 - \frac{u-C}{D-C} & C \leq u \leq D \\ 0 & u \geq D \end{cases} \quad \text{für}$$

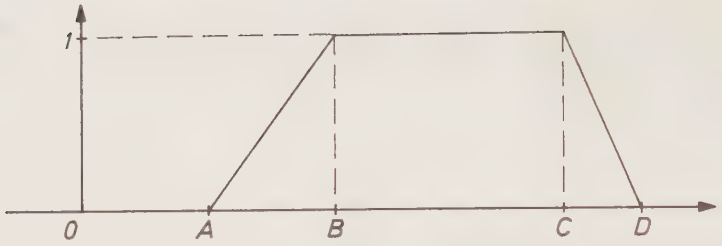


Bild 5.14: Die Fuzzy-Funktion F(u; A, B, C, D)

Auf die durch Kompatibilitätsfunktionen beschriebene Fuzzy-Mengen können auch Mengenoperationen definiert werden. Die Operationen Vereinigung $\cup$, Durchschnitt $\cap$, beschränkte Summe $\oplus$ und beschränkte Differenz $\ominus$ werden folgendermaßen ausgeführt:

$G \cup H := \min(\mu_G(u), \mu_H(u))$

$G \cap H := \max(\mu_G(u), \mu_H(u))$

$G \oplus H := \min(1, \mu_G(u) + \mu_H(u))$

$G \ominus H := \max(0, \mu_G(u) - \mu_H(u))$

Dabei bedeuten + und - die arithmetische Summe bzw. Differenz. Die Negation G' wird definiert als $G' = U \ominus G$.

Beispiel: U = {a,b,c,d,e}

A = {a, 0.5b, 0.5c, 0.7e }

B = {0.7b, 0.4c, 0.2d, e }

$A \cup B$ = {a, 0.7b, 0.5c, 0.2d, e }

$A \cap B$ = {o.5b, 0.4c, 0.7e }

$A \oplus B$ = {a, b, 0.9c, 0.2d, e }

$A \ominus B$ = {a, o.1c }

A' = {o.5b, 0.5c, d, 0.3e }

Als weiteres Beispiel betrachten wir Bild 5.15, das zwei Trapezfunktionen G und H zeigt sowie ihre Vereinigung, Durchschnitt, beschränkte Summe und beschränkte Differenz.

Damit sind Fuzzy-Funktionen informell eingeführt. Ihre Verwendung bei der Definition von Objektklassen wird im nächsten Abschnitt erklärt. In Kapitel 7 wird dann erläutert, wie Fuzzy-Funktionen bei der Bildanalyse zur Bewertung herangezogen werden.

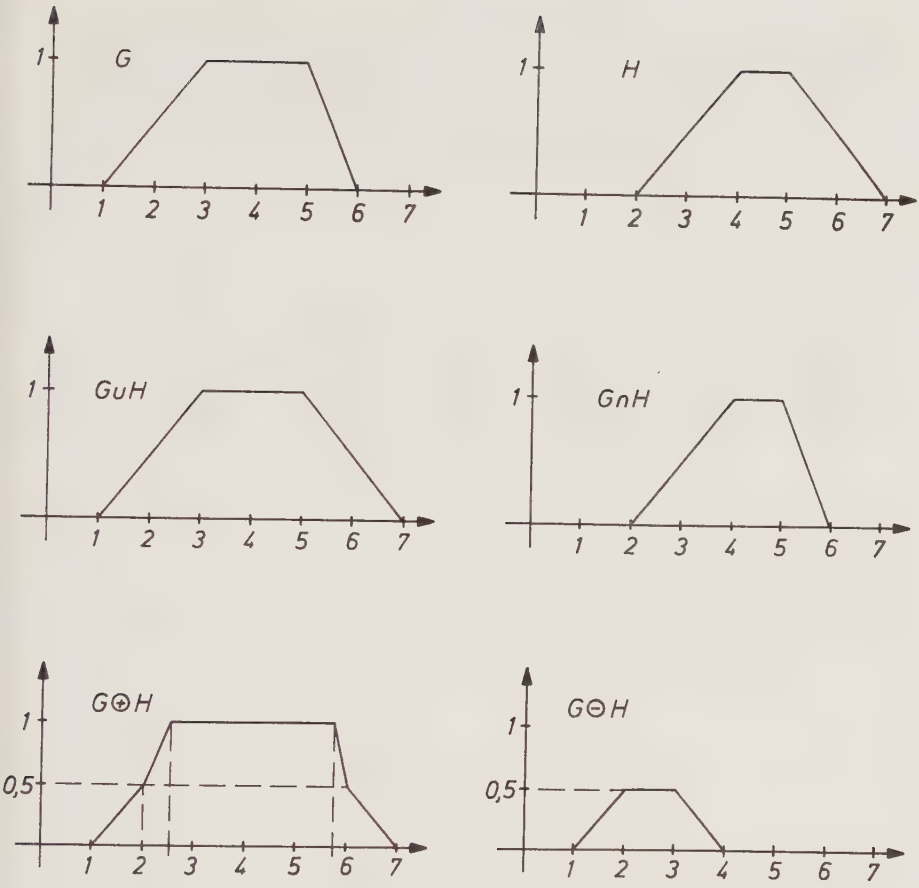


Bild 5.15: Operationen auf Fuzzy-Funktionen

5.4.3 Objektklassenbildung mit Fuzzy-Funktionen

Ziel der Objektklassenbildung sind Objektmodelle, die eine Vielzahl von ähnlichen Objekten umfassen. Dazu werden die Parameter der Grundfläche, der Deckfläche und die Höhe der Primelemente mit Fuzzy-Funktionen definiert. Man erreicht damit Variationen in der Größe und den Ausmaßen der einzelnen Primelemente. Wenn die Objektklassen entsprechend gefaßt sind, ist auch eine Variation der Form möglich. Allerdings gilt natürlich immer die Einschränkung, daß zu umfangreiche Objektklassen zu ungenau und deshalb für die Analyse nicht

hilfreich sind. Was mit der Parametervariation innerhalb einer solchen Objektklasse möglich ist, zeigt beispielhaft Bild 5.16. Bei diesen einfachen Stühlen variiert nur die Lehne. Die Fuzzy-Funktion für die Höhe der Lehne läßt sowohl Variation 1 als auch Variation 2 zu, während für die 3. Variation die Fuzzy-Funktionen der lokalen Koordinaten von Grund- und Deckfläche verantwortlich sind.

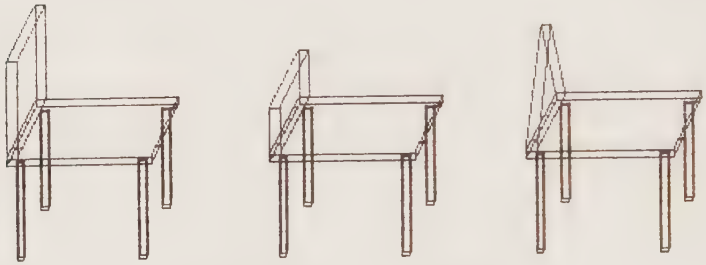


Bild 5.16: Variationen eines Stuhls

Auch die Werte der Oberflächenattribute können mit Fuzzy-Funktionen definiert werden. Damit kann man z.B. Variationen der Intensität ausdrücken mit der Fuzzy-Funktion I (u;A,B,C,D) und den Parametern A = 55, B = 70, C = 90, D = 100.

Neben den Primelementen kann auch die Zusammensetzung der Objekte aus Objektteilen variabel gehalten werden. Die Transformationsparameter, Translation und Rotation, für die Umrechnung der lokalen KOS in das Objekt-KOS können mittels Fuzzy-Funktionen definiert werden. Damit erreicht man, daß die örtlichen Beziehungen der Objektteile zueinander vager beschrieben werden können. Das gleiche gilt für die örtlichen Beziehungen der Objekte innerhalb einer Objektgruppe. Weiterhin ist es möglich, die Anzahl gleicher Objektteile durch die Angabe einer minimalen und einer maximalen Anzahl einzuschränken. Für den Schreibtisch in Bild 5.17 gilt z.B. eine minimale Anzahl von einer Schublade und eine maximale Anzahl von drei Schubladen. Auch die örtliche Lage der Schubladen läßt sich innerhalb des Schreibtisches variieren, wie man auf Bild 5.17c sieht.

Mit dieser Definition von Objektklassen erreichen wir die Beschreibung der in Abschnitt 5.4.1 geforderten Variationsmöglichkeiten. Das Aufstellen der Objektklassen durch die Bestimmung der entsprechenden Fuzzy-Funktion muß bisher

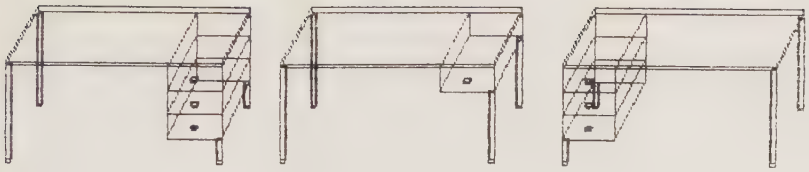


Bild 5.17: Variationen eines Schreibtisches

noch durch einen Bediener erfolgen, und dieser ist meist auf Heuristik und Intuition angewiesen. Spezielle Problembereiche, wie z.B. industrielle Szenen, in denen die Variabilität der Objekte gering ist, lassen an einen automatischen Ansatz zum Lernen dieser Objektklassen denken, bei dem nur noch der Klassenname eingegeben werden muß. Dies wurde hier aber nicht weiter verfolgt.

5.5 Bedingungen

Mit der sehr weitgehenden Variabilität bei Objektklassen erreicht man eine ziemlich freie Definition von Objektmodellen. Um diese wieder einschränken zu können, ist die Angabe von Bedingungen im Modell möglich. Für die Notwendigkeit, Bedingungen zu formulieren, gibt es zwei wesentliche Gründe.

Leitet man aus den Objektmodellen Objekte ab, so ist die Plausibilität weiterhin gewährleistet, sofern die Primelemente plausibel sind. Die gewonnenen Objekte sind zwar physikalisch im mathematischen Sinne zulässig, aber semantisch nicht zu erklären.

Zwei Beispiele sollen dies verdeutlichen. Bild 5.18 zeigt ein Einfamilienhaus, bei dem das Dach einen Meter über dem Haus schwebt. Dies kommt dadurch zustande, daß die Höhe des Hauses in gewissen Grenzen schwanken kann und die Transformation des Daches sich in den gleichen Grenzen bewegt. Da nun beide Parameter unabhängig voneinander sind, könnte ein solches Objekt generiert werden. Als weiteres Beispiel ist in Bild 5.19 ein unproportionierter Schreibtisch dargestellt. Betrachten wir einmal nur die Schreibtischplatte, so kommt dies zustande, weil man sich eine solche Tischplatte rechteckig vorstellt. Dies ist hier nicht beachtet, obwohl der Schreibtisch so aus dem entsprechenden Objektmodell erzeugt werden kann. Während beim zweiten Beispiel schon ein einzelnes Primelement nicht unseren Vorstellungen entsprach, tritt beim ersten Beispiel das Problem erst auf Objektebene auf. Beides sollte mit

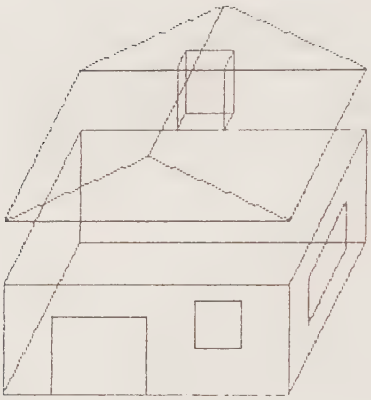


Bild 5.18: Unzulässiges Einfamilienhaus

entsprechenden Bedingungen abgefangen werden können.

Auch für die Bildanalyse sollte sich die Einführung von Bedingungen positiv auswirken. Dabei handelt es sich zum einen um den Teil der Analyse, der die Interpretation von Strukturen betrifft, die im Bild gefunden werden. Wird z.B. eine trapezförmige Region gefunden, so ist eine Interpretation als Schreibtischplatte nur dann möglich, wenn diese nicht laut Bedingung rechtwinklig sein muß. Zum andern kann ein Interpretationsalgorithmus auch Hypothesen aus dem Modell generieren lassen. Berücksichtigt man hierbei Bedingungen, so sinkt die Zahl der möglichen Hypothesen merklich.

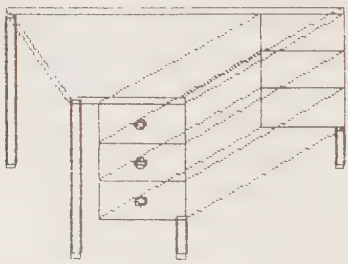


Bild 5.19: Unzulässiger Schreibtisch

Bedingungen sind als Funktionen im Modell vorhanden. Sie liefern als Ergebnis entweder einen logischen Wert (wahr oder falsch) oder eine Wert zwischen 0 und 1, ähnlich den im letzten Abschnitt vorgestellten Fuzzy-Funktionen. Da sie erst bei der Analyse ausgewertet werden können, gehören sie zum prozeduralen Wissen (siehe Kapitel 2). Sie sind über ihre Funktionsnamen an die einzelnen Objektmodelle angebunden, so daß die Bearbeitungen einer solchen Funktion direkt durch den Aufruf dieses Funktionsnamens erfolgen kann, wobei die Versorgung mit Parametern sichergestellt werden muß. Eine Realisierung dieses Mechanismus unter dem Betriebssystem iRMX86 wird im siebten Abschnitt dieses Kapitels erläutert.

Bedingungen werden an Grundflächen, Deckflächen, Primelemente und Objekte angebunden. Über die Namen der jeweiligen Größe erreicht man alle dafür definierten Bedingungen und kann diese durch den Aufruf der entsprechenden Funktionsnamen dann ausführen. Sind mehrere Bedingungen für eine Größe abgelegt, so sind diese, abhängig von der Realisierung (siehe Abschnitt 5.7), implizit mit UND oder ODER verknüpft. Für das Einfamilienhaus aus Bild 5.18 lautet die Bedingung dann: Oberfläche 6 des Hauses und Oberfläche 1 des Daches müssen in Objektkoordinaten identisch sein. Und bei der Schreibtischplatte aus Bild 5.19 muß sowohl die Grundfläche als auch die Deckfläche rechteckig sein.

Da es auch möglich ist, Größen mit gewissen bedingten Eigenschaften, wie z.B. alle rechteckigen Grundflächen, aus dem Modell abzuleiten, unterstützen solche Bedingungen die Analyse merklich. Auch Levine und Shaheen /LEV 81/ verwenden z.B. ausführlich Bedingungen in ihrem Bildanalysesystem, allerdings zur Bildvorverarbeitung. Aber sie beschreiben Bedingungen als ein wichtiges Hilfsmittel bei der Bildanalyse.

5.6 Syntax der Modellbildung

In den bisherigen Abschnitten dieses Kapitels haben wir die Strukturen beschrieben, die in dem BMS für den Schritt der Modellbildung zur Verfügung stehen. Um dies kurz und übersichtlich zusammenzufassen, geben wir eine Syntax dieser Modellbildung in einer Backus-Naur-ähnlichen Form /SCN 75/ an. Sie besteht aus zwei Teilen: einem beschreibenden Teil, in dem Geometrie, Oberflächenattribute und Bedingungen erklärt werden und einem auswertenden Teil, in dem Reihenfolge und Zusammenhang bei der Bewertung von Vergleichen entsprechend dem Explizitmachen impliziter Variabler bei der "test-score-semantics" (vergleiche Abschnit 5.4) erläutert sind. Es folgen jetzt einige Erklärungen zu

beiden Teilen.

Im beschreibenden Teil wird die auf der linken Seite stehende Größe jeweils definiert. Dabei bedeutet ein v auf der rechten Seite alternative Definitionen, während durch ein & auf der rechten Seite verbundene Terme konjunktiv aufgefaßt werden. Ein Punkt "." steht für ein beliebiges Zeichen. Somit kann z.B. FU.. sowohl als FUHÖ als auch als FULI interpretiert werden. Drei Punkte "..." dagegen deuten an, daß die dort beim Aufbau eines Modells stehende Liste erweitert werden kann. Durch ...KORD, also z.B. LOKKORD, sind die zu den jeweiligen Größen gehörenden Koordinatensysteme gekennzeichnet. Insgesamt definiert diese Syntax die Modellbildung zurückgehend bis auf ganze Zahlen, reelle Zahlen oder Funktionsnamen, also Zeichenketten. Diese Terme werden im üblichen Sinn verstanden. Die Bedeutung der einzelnen Größe ist aus der anhängenden Erklärung oder aus dem Zusammenhang von Kapitel 5 ersichtlich.

Wie schon erwähnt, zeigt der auswertende Teil auf, welche Bewertungen erfolgen müssen, um eine gegebene Größe zu bewerten. Am Anfang sind nochmals die beiden im Prinzip gleichen, aber hier unterschiedlich betrachteten Funktionstypen erwähnt. SICH(BED) bedeutet die Auswertung einer bestimmten, durch ihren Namen gekennzeichneten Bedingung unter Berücksichtigung spezieller Parameter PAR1,...,PAR4. Steht dagegen eine andere Größe in der Klammer (SICH(ENTITY)), so ist damit die Fuzzybewertung dieser Größe gemeint, die sowohl durch die direkte Auswertung einer Fuzzy-Funktion, als auch die Verknüpfung mehrerer Einzeltests zustande gekommen sein kann. Soll z.B eine Grundfläche (GRUNDF) mit einer im Bild gefundenen Fläche verglichen werden, so muß man z.B. auf eine Kreisfläche (KREIS) und die zu dieser Grundfläche zählenden Bedingungen (BED) zurückgreifen. Für die Kreisfläche wiederum genügt die Auswertung des Radius. Verknüpft man nun nach der in Kapitel 7. definierten Operation (hier mit § bezeichnet) die Einzelwerte, so erhält man einen Vergleichswert zwischen einer im Bild gefundenen Fläche und einer bestimmten Grundfläche im Modell. In ähnlicher Weise kann diese gesamte Tabelle für die Analyse verwendet werden.

Modellbildung

SZENE ::= (OBJEKT v OBJGRU) & (REGIO)* &
((OBJEKT & ZUSH) v (OBJGRU & ZUSH))$^+$ & SZEKORD

OBJGRU ::= OBJEKT & (OBJEKT & ZUSH)$^+$ & GRUKORD

```
OBJEKT   ::= OBJTEIL & ((OBJEKT & ZUSH) v (OBJTEIL & ZUSH))* &
             BED & OBJKORD

REGIO    ::= (ATTRIB)+ & (BED)*

OBJTEIL  ::= VERZYL & (OBNUM & ATTRIB)+ & (BED)*

VERZYL   ::= GRUNDF & DECKF & HÖHE & FUHÖ & LOKKORD v
             GRUNDF & HÖHE & FUHÖ & SWEEP & LOKKORD

OBNUM    ::= ganze Zahl

GRUNDF   ::= (VIERE v KREIS v DREIE v ... ) & (BED)*

DECKF    ::= (VIERE v KREIS v DREIE v ... ) & (BED)*

SWEEP    ::= FALI & FULI & FAKO & FUKO

VIERE    ::= KOX1 & FUX1 & KOY1 & FUY1 & KOX2 & FUX2 & KOY2 & FUY2 &
             KOX3 & FUX3 & KOY3 & FUY3 & KOX4 & FUX4 & KOY4 & FUY4

DREIE    ::= KOX1 & FUX1 & KOY1 & FUY1 & KOX2 & FUX2 & KOY2 & FUY2 &
             KOX3 & FUX3 & KOY3 & FUY3

KREIS    ::= RADIUS & FURA

ATTRIB   ::= INTENS v FARBE v TEXTUR v ...

INTENS   ::= IWERTI & FUIN

FARBE    ::= IWERTR & FURO & IWERTG & FUGR & IWERTB & FUBL

TEXTUR   ::= RWERT1 & FUT1 & RWERT2 & FUT2 & RWERT3 & FUT3 & RWERT4 & FUT4

ZUSH     ::= TRANSX & FUTX & TRANSY & FUTY & TRANSZ & FUTZ &
             ROTX & FURX & ROTY & FURY & ROTZ & FURZ

FU..     ::= KOW1 & KOW2 & KOW3 & KOW4
```

BED ::= FUNKTION & PAR1 & PAR2 & PAR3 & PAR4

FUNKTION ::= Zeichenkette

PAR. ::= reelle Zahl v ganze Zahl v Zeichenkette

HÖHE ::= reelle Zahl

TRANS. ::= reelle Zahl

ROT. ::= reelle Zahl

IWERT. ::= ganze Zahl

RWERT. ::= reelle Zahl

RADIUS ::= reelle Zahl

KO.. ::= reelle Zahl

FALI ::= reelle Zahl

FAKO ::= reelle Zahl

Modellbildung (Bewertung)

SICH(BED) = BED(PAR1, PAR2, PAR3, PAR4)

SICH(ENTITY) = FUZZ(WERT)

SICH(SZENE) = (SICH(OBJEKT) v SICH(OBJGRU)) § ((SICH(OBJEKT) § SICH(ZUSH)) v (SICH(OBJGRU) § SICH(ZUSH)))$^+$ § (SICH(REGIO))* & (SICH(BED))*

SICH(OBJGRU) = SICH(OBJEKT) § (SICH(OBJEKT) § SICH(ZUSH))$^+$ § (SICH(BED))*

SICH(OBJEKT) = SICH(OBJTEIL) § ((SICH(OBJEKT) § SICH(ZUSH)) v (SICH(OBJTEIL) § SICH(ZUSH)))* § (SICH(BED))*

```
SICH(REGIO)   = (SICH(ATTRIB))+ § (SICH(BED))*

SICH(OBJTEIL) = SICH(VERZYL) § (SICH(ATTRIB))+

SICH(VERZYL)  = SICH(GRUNDF) § SICH(DECKF) § SICH(HÖHE) v
                SICH(GRUNDF) § SICH(HÖHE) § SICH(SWEEP) § (SICH(BED))*

SICH(GRUNDF)  = (SICH(VIERE) v SICH(KREIS) v SICH(DREIE) ... ) § (SICH(BED))*

SICH(DECKF)   = (SICH(VIERE) v SICH(KREIS) v SICH(DREIE) ... ) § (SICH(BED))*

SICH(SWEEP)   = SICH(FALI) § SICH(FAKO)

SICH(VIERE)   = SICH(KOX1) § SICH(KOY1) § SICH(KOX2) § SICH(KOY2) §
                SICH(KOX3) § SICH(KOY3) § SICH(KOX4) § SICH(KOY4)

SICH(DREIE)   = SICH(KOX1) § SICH(KOY1) § SICH(KOX2) § SICH(KOY2) §
                SICH(KOX3) § SICH(KOY3)

SICH(KREIS)   = SICH(RADIUS)

SICH(ATTRIB)  = SICH(INTENS) v SICH(FARBE) v SICH(TEXTUR) v ...

SICH(INTENS)  = SICH(IWERTI)

SICH(FARBE)   = SICH(IWERTR) § SICH(IWERTG) § SICH(IWERTB)

SICH(TEXTUR)  = SICH(RWERT1) § SICH(RWERT2) § SICH(RWERT3) § SICH(RWERT4)

SICH(ZUSH)    = SICH(TRANSX) § SICH(TRANSY) § SICH(TRANSZ) §
                SICH(ROTX) § SICH(ROTY) § SICH(ROTZ)
```

Erklärung:

v	alternativ
&	und
(**)+	beliebig oft, aber mindestens einmal
(**)*	beliebig oft, kann auch wegfallen

::=	definiert durch
=	ergibt sich aus
§	Verknüpfungsfunktion für die Bewertung
ATTRIB	Name eines Oberflächenattributs
BED	Bedingung
DECKF	Deckfläche
ENTITY	Größe, für die eine Fuzzy-Funktion definiert ist
FAKO	konstanter Faktor der "sweeping rule"
FALI	linearer Faktor der "sweeping rule"
FARBE	Farbe
FU..	Fuzzy-Funktion
FUNKTION	auswertbare Funktion
GRUNDF	Grundfläche
HÖHE	Höhe
INTENS	Intensität
IWERT.	ganze Zahl
KO..	Koordinate
KREIS	Kreis
OBJEKT	Objekt
OBJGRU	Objektgruppe
OBJTEIL	Objektteil
OBNUM	Oberflächennummer
PAR.	Parameter
RADIUS	Radius
REGIO	Region
ROT.	Rotation um die x-Achse (bzw. y-Achse oder z-Achse)
RWERT.	reelle Zahl
SICH	Sicherheit
SWEEP	sweeping rule
SZENE	gesamte Szene
TEXTUR	Textur
TRANS.	Translation in x- bzw. y- oder z-Richtung
VERZYL	Verallgemeinerter Zylinder
VIERE	Viereck
WERT	aktueller Wert
ZUSH	Zusammenhang
...KORD	zugehöriges Koordinatensystem

5.7 Darstellung auf dem Rechner

Wie am Anfang dieses Kapitels mit Bild 5.1 erläutert wurde, umfaßt ein BMS zwei Teile. Die Modellbildung wurde ausführlich dargelegt. Dieser Abschnitt befaßt sich nun mit dem zweiten Teil, der Darstellung am Rechner. Dabei sollen die "mathematischen" Modelle (wie in Kapitel 4 ausgeführt wurde) so auf dem Rechner gespeichert werden, daß auf sie geeignet und schnell zugegriffen werden kann. Als Formalismus für die Wissensdarstellung wurde hier eine relationale Datenbank gewählt. Da zum Zeitpunkt des Beginns der Arbeit auf dem Rechner intel SYS 86/330 unter dem vorhandenen Betriebssystem iRMX86 keine relationale Datenbank verfügbar war, wurde eine eigene entwickelt und realisiert. Diese relationale Datenbank wird im Abschnitt 5.7.1 vorgestellt. Danach wird dann gezeigt, mit welchen Relationen das Modell auf dem Rechner abgelegt wird.

5.7.1 Verwendete relationale Datenbank

Wir implementierten eine Standarddatenbank nach dem relationalen Modell. Dabei wurden u.a. folgende Entwurfskriterien und Randbedingungen berücksichtigt:

- Ein-Benutzer-Betrieb
- Wichtig ist die Programmschnittstelle, weniger die Interaktion mit dem Benutzer
- Abfragesprache relationale Algebra
- Modularität
- Einfache Zugriffsrechte
- Schneller Zugriff
- Relationen der 1. Normalform
- Primärschlüssel und Sekundärschlüssel
- Portabilität

Um eine möglichst große Modularität zu erreichen, wurde als Grundlage ein Schichtenmodell nach Härder /HAE 78/ gewählt (siehe Bild 5.20). Das Datensystem muß die Verwaltungs- und Zugriffsanfragen, die über die Programmschnittstelle oder vom Benutzer kommen, auflösen in Ein-Tupel-Anfragen, die an das Zugriffssystem weitergegeben werden. Dieses berechnet explizit das Tupel, das dann vom Speichersystem aus der Relation gelesen wird. Dabei muß gewährleistet sein, daß jede Schicht nur auf die direkt darunterliegende zugreifen darf. Benutzer- bzw. Programmschnittstelle und Datensystem sind bei Fichte /FIC 85/, Zugriffssystem und Speichersystem bei Schott /SCO 85/ beschrieben. Die Realisierung erfolgte zum überwiegenden Teil in FORTRAN 86, dem auf den

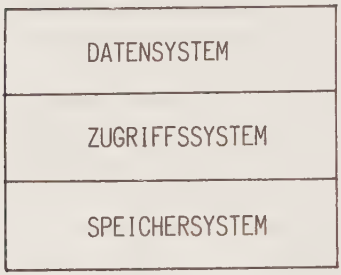


Bild 5.20: Schichtenmodell für Datenbanksysteme

Mikroprozessor intel 8086 angepaßten Dialekt von FORTRAN 77. Lediglich das Speichersystem ist in PLM 86 geschrieben, da für die Dateiverwaltung Betriebssystemaufrufe verwendet werden, die in FORTRAN 86 nicht einzubinden sind. So ist einerseits eine höhere Zugriffsgeschwindigkeit beim Dateizugriff erreicht, aber andererseits durch die weitgehende Verwendung von FORTRAN 86 die Forderung nach Portabilität erfüllt.

Im Datensystem ist eine Unterscheidung in Administrator- und Benutzerroutinen möglich. Zu den Administratorroutinen zählen die Verwaltung des Benutzerkontrollblocks, die Verwaltung der Relationen, Statusabfragen u.ä. Der Benutzerkontrollblock beinhaltet den Benutzernamen, ein Paßwort, das zur Sicherung vor unerlaubter Benutzung der Datenbank eingesetzt werden kann, und eine Liste der Relationen, auf die dieser Benutzer keine Zugriffsberechtigung hat. Damit ist die Erteilung von Zugriffsrechten nur auf Relationen - und nicht auf Tupel- oder Attributebene möglich (vgl. /HUL 82/). Ein einfacher Vergleich der Relationsnamen, der allerdings bei jedem Zugriff erfolgen muß, genügt dann, um die Zugriffsberechtigung festzustellen. Weiterhin gehört die Definition, Änderung und Löschung von Relationen zu den Administratorroutinen. Jeder Relation ist ein Relationenkontrollblock zugeordnet, der folgende Informationen enthält:

- Relationsname
- Tupelanzahl
- maximale Attributlänge
- aktuelle Attributanzahl

- maximale Anzahl von Wertebereichseinschränkungen
- Attributnamen
- Attributformate
- Attributschlüssel
- Explizite Wertebereichseinschränkungen
- Relationsbeschreibung (maximal 250 Zeichen)

Dabei sind als Attributformate INTEGER, REAL, CHARACTER und FUNCTION, das wie CHARACTER behandelt wird, zugelassen. Als Attributschlüssel ist sowohl die Primärschlüsseleigenschaft als auch die Definition als Sekundärschlüssel möglich. Wertebereichseinschränkung und Relationsbeschreibung sind optional. Zuletzt zählen auch Statusausgaben, wie Relationskontrollblöcke oder Relationennamenkatalog, zu den Administratorroutinen.

Zu den Benutzerroutinen gehören die Operationen der Abfragesprache, Tupelmanipulationen und Sortieralgorithmen. Als Abfragesprache wurde die relationale Algebra ausgewählt. Von den dort vorhandenen Relationenoperationen sind Projektion, Selektion und Verbund implementiert, wobei mehrere Verbundarten (Gleich-Verbund, Größer-Verbund, Kleiner-Verbund, Größer-Gleich-Verbund, Kleiner-Gleich-Verbund und Ungleich-Verbund) zulässig sind. Als Mengenoperationen gibt es Vereinigung, Durchschnitt und Differenz. Somit fehlen nur Division und kartesisches Produkt. Im Vergleich dazu hat Bense /BEE 80/ Projektion, Selektion, Verbund, Vereinigung und Differenz implementiert, während bei Hülsmeier/Hülsmeier /HUL 82/ nur Projektion, Selektion, Vereinigung und Differenz zum Sprachumfang gehören. Nur zum Zwecke einer geeigneten Ausgabe kann eine Menge von Tupel anhand eines Attributs in aufsteigender oder absteigender Folge sortiert werden. Dabei gilt für Zeichenketten die lexikalische Ordnung. Bei der Manipulation von Tupeln ist Eingabe, Änderung und Löschung berücksichtigt, wobei die Erhaltung der Konsistenz der Schlüsselinformationen bei diesen Operationen gewährleistet sein muß.

Hat das Datensystem eine Anfrage aufgelöst, so stellt das Zugriffssystem Strukturen zur Verfügung, um schnell das oder die gesuchten Tupel zu finden. Ein gefundenes Tupel wird dabei durch den Tupelindentifikator TID gekennzeichnet, der in jeder Relation eindeutig ist. Das Speichersystem muß anhand einer Tabelle den Tupelidentifikator in die Recordnummer der zu dieser Relation gehörenden Datei umrechnen und kann dann auf diesen Record zugreifen. Da die Tupel in einer Relation ungeordnet gespeichert sind, muß das Zugriffssystem für eine Abfrage die gesamte Relation durchsuchen. Dies heißt Relationenscan und

ist sehr zeitaufwendig. Deshalb ist im Zugriffssystem noch eine schnellere Zugriffsmöglichkeit realisiert, die über den Primärschlüssel geht und Indexscan heißt. Dazu müssen zusätzlich Tabellen im System gehalten werden, in denen die Werte der Primärschlüsselattribute (maximal 5) geordnet niedergelegt werden. Prinzipiell unterscheidet man hier die Möglichkeiten der Schlüsseltransformation und des Schlüsselvergleichs /WED 76/. Die Schlüsseltransformation, bei der mittels einer Hash-Transformation jedem Schlüssel ein Speicherplatz zugeordnet wird, ist bei konfliktfreiem Zugriff optimal bezüglich der Zugriffszeit. Da aber bei unserer Anwendung wenig über die Schlüsselbereiche ausgesagt werden kann und deshalb viele Transformationskonflikte zu erwarten sind, wurde ein baumstrukturierter Schlüsselvergleich mit blattorientierten Mehrwegbäumen, sog. B^*-Bäumen, gewählt. Hier seien nochmals die wichtigsten Eigenschaften eines B^*-Baums aufgeführt:

- Jeder Knoten hat wenigstens k + 1 Söhne (Ausnahme: Wurzelknoten und Blattknoten
- Jeder Knoten besitzt höchstens 2k + 1 Söhne
- Der Wurzelknoten ist entweder ein Blattknoten oder er hat wenigstens zwei Söhne
- Jeder Pfad von der Wurzel bis zu einem Blattknoten hat dieselbe Länge h.

Damit bestimmt der Parameter h die Höhe des Baumes und da nur in den Blättern die Daten stehen, auch die Anzahl der Zugriffe auf den Baum bis zum Auffinden eines Schlüssels und des zugehörigen TID. Knoten, die keine Blattknoten sind, haben dann als Einträge Zeiger P_i und Primärschlüsselwerte K_i. Blattknoten beginnen mit einer Marke M, die sie als Blattknoten identifiziert und beinhalten dahinter dann Paare mit jeweils einem Primärschlüsselwert und dem zugehörigen Tupelidentifikator TID (siehe Bild 5.21). Für Schlüssel K_i, die nicht in einem Blattknoten liegen, findet man den TID, indem man den zugehörigen Zeiger K_i auf die nächste Ebene verfolgt und von dort jeweils mit dem Zeiger K_0 weitergeht, bis man einen Blattknoten erreicht, dessen Eintrag TID_0 der gesuchte ist. Selbst bei kleinen Relationen von 30 bis 90 Tupel erzielt man damit eine Zeitreduzierung um den Faktor 3 bis 7 gegenüber dem Relationenscan (vergleiche Tabelle 5.1 /SCO 85/). Daneben gibt es auch die Möglichkeit, Sekundärschlüsselattribute zu definieren und einen schnellen Zugriff mittels eines geeigneten Indexscan darüber abzuwickeln (siehe /SCO 85/).

A) KNOTEN: $P_0\ K_1\ P_1\ K_2\ P_2\ \ldots\ K_{2K}\ P_{2K}$

B) BLATTKNOTEN: $M\ TID_0\ K_1\ TID_1\ K_2\ TID_2\ \ldots\ K_{2K}\ TID_{2K}$

C) B^*-BAUM

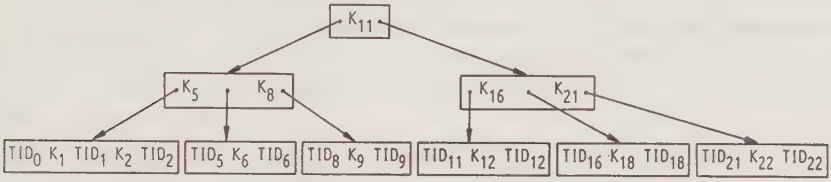


Bild 5.21: Datenstrukturen des Zugriffssystems

TUPELANZAHL	ZEIT FÜR RELATIONEN-SCAN	ZEIT FÜR INDEX-SCAN
90	50	6
65	46	6
46	28	6
32	19	6

Tabelle 5.1: Abfragezeiten in Sekunden

5.7.2 Darstellung des Modells durch Relationen

Das realisierte Modell besteht zur Zeit aus 15 Relationen. Die Abhängigkeiten der Relationen sieht man in Bild 5.22, eine Auflistung der Attribute findet sich im Anhang A. Entsprechend der Syntax aus Abschnitt 5.6 legen die Relationen einzelne der dortigen Zeilen fest. Die Endung K der Relationennamen soll ausdrücken, daß es sich bei diesen Relationen um Klassen handelt, d.h. die in Abschnitt 5.4 aufgeführten Parameter sind jeweils mit Fuzzy-Funktionen definiert. Eine Szene (SZENE_K) besteht also aus Objektgruppen (OBJGRUPPE_K),

Einzelobjekten (EINZELOBJK) und Regionen (REGIO_K). Gleichartige Objektteile, die jeweils einen verallgemeinerten Zylinder darstellen, sind in OBJTEILE K erfaßt. Die Unterscheidung dieser Objektteile, die in der örtlichen Lage besteht, wird in TEILE_K ausgedrückt. Für die Definition der verallgemeinerten Zylinder (VERALLZYLK) benötigt man die Höhe, die in VERALLZYLK steht, sowie Grund- und Deckfläche (GRUNDFLAEK) oder die "sweeping-rule"(SWEEPING_K). Die Transformationen der verschiedenen Koordinatensysteme ineinander, die durch Translation und Rotation im dreidimensionalen Raum ausgedrückt werden, sind in der Relation ZUSHANG_K zusammengefaßt.

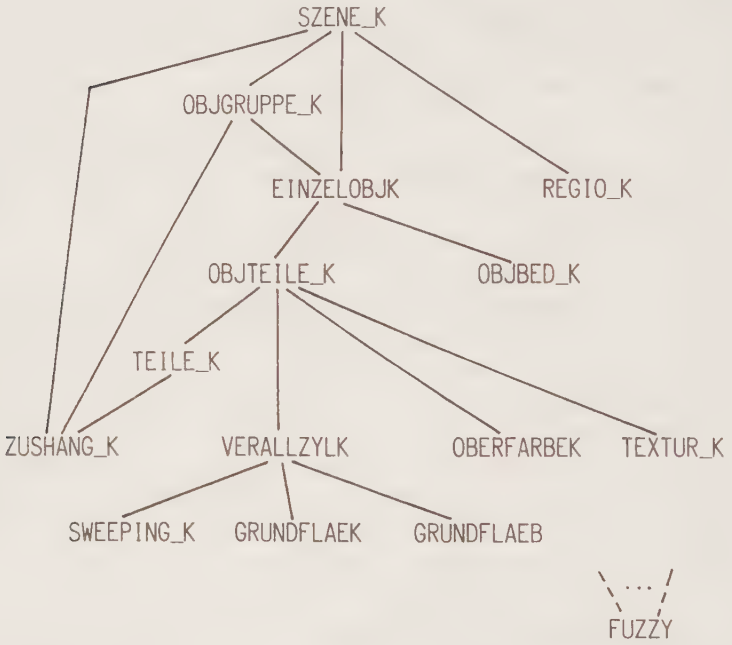


Bild 5.22: Aufbau der Relationen des Modells

Um bei den einzelnen Relationen nicht zu viele Attribute zu haben, sind für die Bedingungen zumindest teilweise eigene Relationen (GRUNDFLAEB, OBJBED K) vorhanden. In die Relation VERALLZYLK sind die Bedingungen aufgenommen. Durch die Einführung des Datentyps FUNCTION für Attribute sind diese Bedingungen speziell gekennzeichnet. Bei der Analyse kann der Wert eines solchen Attributs

direkt als Funktionsname verwendet werden. Man muß dann eine Möglichkeit vorsehen, daß die Prozedur, auf die zugegriffen wird, gestartet wird und die folgenden Attributwerte als Parameter dieser Prozedur behandelt werden. (Das geht, da Attribute in Relationen geordnet sind.) Unter iRMX86 kann man dazu den Befehl "dq$send$command" verwenden. Damit ergibt sich folgende Vorgehensweise:

1. Lies Name der Bedingung.
2. Lies Parameter zu dieser Bedingung.
3. Rufe "dq$send$command" mit der Kommandozeile, die aus dem Bedingungsnamen und den Parametern besteht.

Die zugehörige Prozedur muß die Kommandozeile dann interpretieren mit dem Befehl "dqgetargument".

Die Oberflächenattribute sind in der Relation OBJTEILE_K an die Flächen der VZ gebunden (siehe Abschnitt 5.3). Die Definition der zulässigen Werte erfolgt für die Intensität direkt in OBJTEILE_K, für alle anderen Oberflächenattribute mit eigenen Relationen (OBERFARBEK, TEXTUR_K) über einzelne Attribute, die Namen enthalten. Auch hier wird das Ziel verfolgt, die Anzahl der Attribute in einer Relation nicht zu groß werden zu lassen. Damit kann man bei der Analyse gezielt auf das Modell zugreifen mit Abfragen der Art: Gib mir alle Oberflächen mit der Intensität 50 oder einem bestimmten Texturparameter. Für Abfragen nach der Intensität benötigt man einen, bei allen anderen Oberflächenattributen zwei Zugriffe. Dies erfüllt die Forderung nach einer schnellen Zugriffsmöglichkeit auf Oberflächenattribute.

6. Ein anderer Ansatz für ein Bildanalysemodellierungssystem

Im Rahmen einer Diplomarbeit /GOL 85/ wurden Untersuchungen über die Eignung verschiedener Verfahren zur geometrischen Modellbildung für ein BMS angestellt (siehe Kapitel 3). Als Ergebnis daraus wurde ein alternativer Ansatz für ein BMS definiert /HES 85/. Dieser Ansatz unterscheidet sich aber nur bei der Modellbildung von dem in Kapitel 5 vorgestellten BMS. Dieses Kapitel beinhaltet eine Darlegung dieser Methode und bringt einen Vergleich der beiden BMS.

6.1 Modellbildung

Auch hier stützt sich die Modellbildung auf Primelemente, die zu Objekten zusammengesetzt werden. Dabei werden zwei bekannte Verfahren kombiniert, um die Vorteile beider geometrischer Modellierer auszunutzen. Die Primelemente werden durch Oberflächenzerlegung in einer modifizierten Form beschrieben, während Objekte mittels eines eingeschränkten CSG-Verfahrens gebildet werden.

Wie in Kapitel 3 gezeigt, werden die Primelemente durch ihre Oberflächen, die Oberflächen durch ihre Kanten und die Kanten durch die Eckpunkte in 3D-Koordinaten definiert. Mit diesen Größen läßt sich dann ein Netz (siehe Bild 3.9) aufbauen. Um Flächen gleicher Form zu erhalten, wird für die Oberflächen eine Dreieckszerlegung vorgenommen. Zur weiteren Verarbeitung wird eine Trennung der reinen Geometrie der Primelemente von der Beschreibung der Oberflächen vorgenommen. Dies wird durch eine Aufspaltung des Netzes, das das Primelement beschreibt, in zwei Teilnetze erreicht. Der Inhalt des ersten Teilnetzes erzeugt ein Drahtgittermodell des Primelements, und mit dem zweiten Teilnetz erhält man eine Beschreibung der Oberflächen, die dann auch mit entsprechenden Oberflächenattributen (vgl. Abschnitt 5.3) versehen werden können. Den Aufbau der Teilnetze für einen Keil als Primelement zeigt Bild 6.1. Hierbei fällt auf, daß die durch die Dreieckszerlegung auftretenden "Pseudo"-Kanten nicht zu Verwirrung bei der Analyse führen können. Lediglich in Teilnetz 1 sind die Flächen jeweils als Dreiecke abgelegt.

Außerdem werden die Primelemente parametrisiert, d.h. man führt Parameter ein, die eine zusätzliche Bestimmung ermöglichen. Durchgeführt wurde dies mit den Parametern Länge L, Breite B und Höhe H, die dann die Abschnitte in der lokalen x-, y- bzw. z-Richtung definieren (siehe Bild 6.2). Mit diesem Parameter wird eine objektweite Bindung verschiedener Größen erreicht. Dadurch kann die zusätzliche Definition von Bedingungen entfallen. Ein Problem wie in Bild 5.18

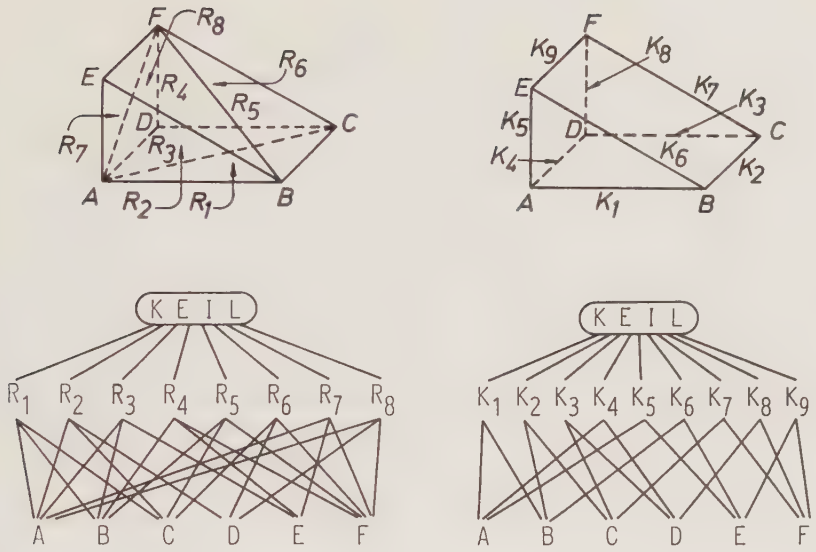


Bild 6.1: Teilnetze für einen Keil als Primelement

kann nicht mehr auftreten, da die Höhe des Hauses durch den gleichen Parameter definiert wird wie die Translation des Daches in z-Richtung, so daß das Dach immer direkt auf dem Haus zu liegen kommt.

Die Objektklassenbildung erfolgt ebenfalls über Fuzzy-Funktionen. Dabei wird der gleiche Typ F(u;A,B,C,D) verwendet wie in Abschnitt 5.4, der mit vier Parametern A, B, C und D beschrieben wird. Mittels dieser Fuzzy-Funktionen werden die oben erklärten Parameter der Primelemente definiert. Damit wird eine

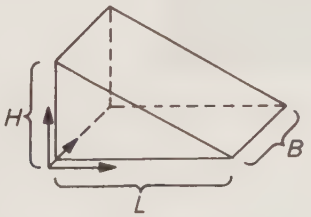


Bild 6.2: Parametrisierung des Primelements Keil

Variation der Ausmaße erreicht.

Zum Aufbau von Objekten aus Primelementen wird ein reduzierter CSG-Baum verwendet. Als Operator ist nur die Vereinigung zugelassen, deshalb wird der CSG-Baum reduziert genannt. Dies erkärt sich aus der Tatsache, daß durch die Oberflächenbeschreibung der Primelemente über die Volumen dieser Element keine Aussage gemacht werden kann. Die Definition von Durchschnitt und Differenz bereitet dann Schwierigkeiten, die umgangen werden sollen. Auch auf die Regularisierung kann damit verzichtet werden. Neben den Vereinigungsknoten gibt es im CSG-Baum noch Bewegungsknoten, die die örtliche Lage eines Teilbaumes oder eines Primelements beschreiben. Zur Beschreibung der örtlichen Lage werden (4 x 4)-Matrizen genommen, die die entsprechenden Transformationen in homogenen Koordinaten enthalten. An dem Beispiel eines Satteldachhauses soll dieser CSG-Baum verdeutlicht werden. Ein Drahtgittermodell des Satteldachhauses beinhaltet Bild 6.3. Den zugehörigen CSG-Baum mit drei Primelementen zeigt Bild 6.4. Dort steht M für Bewegungsknoten und U^* für Vereinigungsknoten.

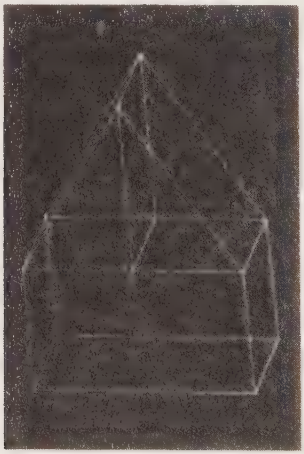


Bild 6.3: Drahtgittermodell eines Satteldachhauses

6.2. Darstellung auf dem Rechner

Zur Darstellung der Objektmodelle auf dem Rechner wird die in Abschnitt 5.7 vorgestellte relationale Datenbank verwendet. Für die Abspeicherung der Objekt-

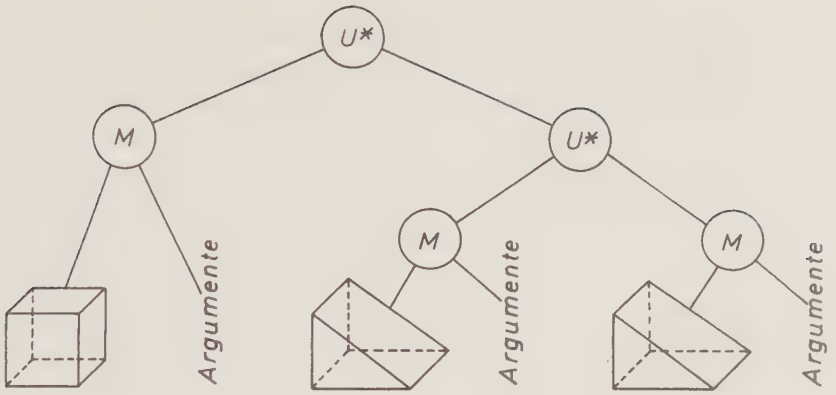
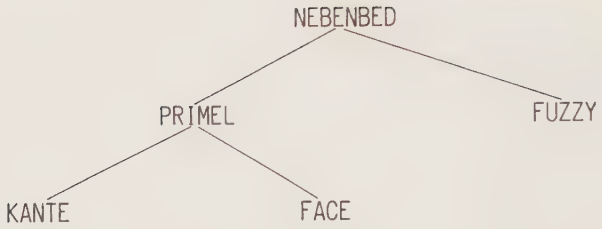


Bild 6.4: CSG-Baum eines Satteldachhauses

modelle sind die in Bild 6.5 aufgeführten Relationen angelegt worden. In der Relation PRIMEL werden die Primelemente anhand ihrer Eckpunkte definiert, wobei hier auch implizit das lokale Koordinatensystem des jeweiligen Primelements festgelegt wird. Die Trennung in zwei Teilnetze bei der Modellbildung findet in zwei verschiedenen Relationen ihren Niederschlag. Während mit der Relation KANTE ein Drahtgittermodell des Primelements beschrieben wird, ermöglicht es die Relation FACE, Oberflächenattribute zu definieren über die Attribute EIGENSCHAFT und WERT. Der Aufbau des CSG-Baums erfolgt in der Relation NEBENBED. Dort sind auch die Parameter der Primelemente aufgeführt und können gegebenenfalls durch Fuzzy-Funktionen spezifiziert werden. Außerdem können diese Parameter auch als variable Werte für die Attribute TRANSLAT-. und ROTATION-. Verwendung finden.

6.3 Vergleich der beiden Ansätze

In diesem Abschnitt sollen die beiden in Kapitel 5 und 6 vorgestellten Ansätze verglichen werden. Dabei ist vor allem die Modellbildung zu betrachten, da diese sich bei beiden Ansätzen unterscheidet. Die Vorteile der Modellbildung mit einem hybriden CSG-Oberflächenverfahren liegen zum einen in der Parametrisierung der Primelemente. Diese Parameter vereinfachen die Objektmodelle, da sie praktisch Größen aus verschiedenen Primelementen aneinanderbinden. Auch innerhalb eines Primelements können damit in begrenztem Umfang Einschränkungen definiert werden. Dadurch erübrigt sich die Definition von Bedingungen für den



RELATION FACE

P-ELEMENT	PUNKT-1	PUNKT-2	PUNKT-3	EIGENSCHFT	WERT

RELATION FUZZY

DUMMY	FUNKTION	PUNKT1	PUNKT2	PUNKT3	PUNKT4

RELATION KANTE

P-ELEMENT	PUNKT-1	PUNKT-2	PUNKTEANZ

RELATION NEBENBED

OBJEKTKLA	OBJEKT	TEIL	LFDNR	LAENGE	FUZZY-L
HOEHE	FUZZY-H	BREITE	FUZZY-B	ROTATION-X	ROTATION-Y
ROTATION-Z	TRANSLAT-X	TRANSLAT-Y	TRANSLAT-Z		

RELATION PRIMEL

P-ELEMENT	ECKPUNKT	PUNKTANZ	X-KOORD	Y-KOORD	Z-KOORD

Bild 6.5: a) Abhängigkeit der Relationen
b) Attribute der Relationen

Ausschluß von semantisch nicht zulässigen Objektmodellen. Zum anderen sind bei dem Verfahren aus Kapitel 6 die Oberflächen im Modell explizit abgespeichert. Es ist damit nicht nötig, diese gegebenenfalls während der Analyse zu berechnen. Damit verlegt man Rechenzeit von der häufig ausgeführten Analyse auf die nur anfangs notwendige Modellbildung. Ähnliches gilt für die bei diesem Verfahren explizit im Modell definierte Vereinigung von Primelementen zu Objektmodellen.

Andererseits sind VZ als Primelemente, mit der Möglichkeit zylindrische Oberflächen zu definieren, allgemeiner als der Eckenansatz aus Kapitel 6, der nur planare Oberflächen zuläßt. Auch die Klassenbildung mit Fuzzy-Funktionen für viele Variable eines Objektmodells und der variablen Anzahl gleichartiger

Objektteile sind Vorteile des Ansatzes mit VZ gegenüber der beschränkten Klassenbildung beim zuletzt vorgestellten Ansatz. Die Bedingungen, die zwar zusätzlichen Aufwand bei der Modellbildung bedeuten, erweisen sich aber, wie in Abschnitt 5.5 erläutert, in mancher Hinsicht als geeignetes Hilfsmittel bei der Analyse.

Im gesamten Vergleich zeigen beide Ansätze einige Vorteile, die schwierig gegeneinander aufzurechnen sind. Zylindrische Oberflächen sind z.B. in manchen

	Modellbildung mit	
	CSG-Oberflächen-Hybrid	Verallgemeinerten Zylindern
Prim-elemente	- Eckenansatz - Oberflächen sind explizit definiert - Parametrisierbar, aber Extraktion der Parameter aus dem Bild schwierig	- allgemeinere Primelemente - Oberflächen müssen berechnet werden - durch Bedingungen eingeschränkt
Objekt-modelle	- explizite Vereinigung - Parameter vereinfachen den Modellierungsaufwand - Klassenbildung nur über Fuzzy-Funktionen - Änderungen nur global ausführbar	- implizite Vereinigung - Zusätzliche Bedingungen nötig, die aber weitreichender sind als die Parameter - Klassenbildung über Fuzzy-Funktionen und gleichartige Teile - Änderungen lokal ausführbar

Tabelle 6.1: Vergleich der beiden Modellierungsverfahren

Problemkreisen sehr wichtig, und eine Approximation mit planaren Dreiecken ist aufwendig, wenn sie notwendig erscheint. Im weiteren wird die Modellbildung auf der Grundlage von VZ verwendet, da sie sich einfacher darstellt, detaillierter ausgearbeitet ist und das hybride CSG-Oberflächenverfahren keine zwingenden Vorteile bringt. Trotzdem könnte in einer zukünftigen Verbesserung des BMS die eine oder andere Idee daraus, z.B. die Parametrisierung, übernommen werden.

7. Zur Analyse von Bildern

Die Analyse eines Bildes läuft, wie in Kapitel 2 bereits besprochen, über mehrere Stufen ab. Nach der Eingabe werden durch die Bildvorverarbeitung aus dem Bild Merkmale und/oder primitive Komponenten extrahiert, die dann interpretiert oder analysiert werden. Dieser Interpretationsprozeß besteht aus einer oder mehreren Bewertungsfunktionen, die die Übereinstimmung zwischen den einzelnen Komponenten aus Bild und Modell bewerten, und einer Vorgehensstrategie, die die Auswahl trifft, welche Komponenten in welcher Reihenfolge verglichen werden. Diese Vorgehensstrategie ist Teil des Kontrollmoduls aus Bild 2.2, während die Bewertungsfunktionen entweder lokal für die einzelnen Komponenten im Modell gespeichert sind oder global für das ganze System zum Kontrollmodul gehören. Es gibt für dieses Problem verschiedene Ansätze, von denen wir einige kurz erläutern wollen, um unseren Ansatz einordnen zu können.

Auf Rosenfeld u.a. gehen die sog. Relaxationsverfahren zurück (vgl. /ROS 76, WAN 83/). Grundlage dabei sind symbolische Beschreibungen von Bild und Modell auf der Basis von Eigenschaften wie Kanten, Regionen usw. In einem ersten Schritt wird eine initiale Zuordnung Bild - Modell erstellt. Nun benötigt man Kompatibilitätsfunktionen, die aufgrund statistischen oder heuristischen Wissens die Verträglichkeit der lokalen Zuordnung der Komponenten innerhalb einer begrenzten Umgebung bewerten. In einem iterativen Prozeß wird nun versucht, die Zuordnung mithilfe der Kompatibilitätsfunktionen zu verbessern bis einige Eigenschaften mit hoher Sicherheit zugeordnet werden können. Mit diesem Wissen wird eine neue Anfangszuordnung berechnet und der Prozeß beginnt von neuem. Bild 7.1 faßt diese Vorgehensweise noch einmal zusammen. Price /PRI 85/ gibt einen Vergleich von verschiedenen Relaxationsverfahren aufgrund von Fehlern und Geschwindigkeit der Konvergenz. Dabei schneiden die Verfahren von Faugeras und Price /PRI 85/ sowie Hummel und Zucker /HUM 83/ am besten ab. Zur Wissensdarstellung verwendet Price ein semantisches Netz.

Als Grundlage für den Interpretationsprozeß sind relationale Strukturen weit verbreitet /SHP 84/. Eine relationale Beschreibung D_X ist eine Folge von Relationen $D_X=\{R_1,...R_q\}$, wobei für alle $i=1,...,q$ gilt: R_i ist eine t_i-stellige Relation. Jede Relation R_i ist wiederum eine Teilmenge von X^{ti}, $R_i \subseteq X^{ti}$. Relationale Strukturen beinhalten nach Radig /RAD 84/ noch zusätzlich die Grundmenge X, also $RS=\{X, R_i,...,R_q\}$. Diese Grundmenge X entspricht nun den oben erwähnten primitiven Komponenten (wie Kanten, Regionen usw.). Hat man

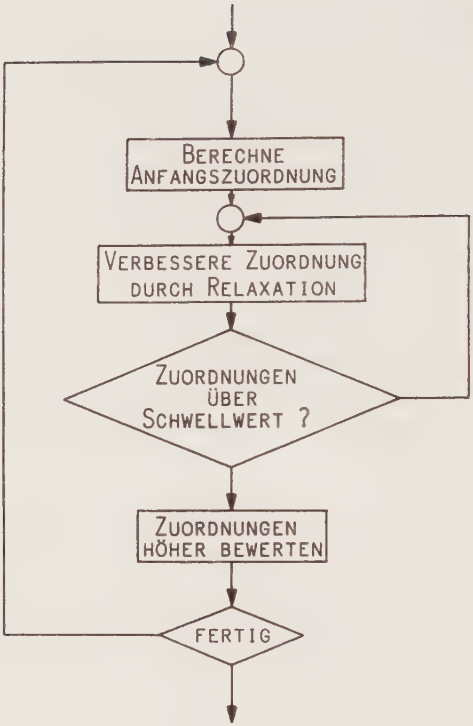


Bild 7.1: Grundalgorithmus für Relaxationsverfahren

nun eine relationale Beschreibung $D_A=\{R_1,...,R_q\}$ für das Bild und eine relationale Beschreibung $D_B=\{S_1,...,S_q\}$ für das zugehörige Modell gegeben, so liegt das Ziel des Interpretationsprozesses darin, eine möglichst gute Abbildung zwischen D_A und D_B zu finden. Da aus den bekannten Gründen Bild und Modell nicht übereinstimmen, ist es auch nicht möglich, eine bijektive Abbildung zwischen D_A und D_B zu finden. Radig /RAD 84/ sowie Cheng und Huang /CHE 81/ führen dieses Problem deshalb auf einen Vergleich von Graphen zurück. Damit muß man maximale Teilgraphen (sog. "cliques") in D_A und D_B finden, die isomorph sind.

Einen anderen Ansatz wählen Shapiro und Haralick /SHP 85/, die ein relationales Abstandsmaß $GD(D_A,D_B)$ definieren und nachweisen, daß $GD(D_A,D_B)$

eine Metrik ist. Damit ist für den kleinsten Wert von $GD(D_A,D_B)$ die beste Interpretation erreicht. Es gilt:

$$GD(D_A,D_B) = \min_{f: A \to B} E(f).$$

Man listet also alle möglichen bijektiven Abbildungen f der Grundmengen A und B ineinander auf und berechnet für jede solche Abbildung f den Fehler, der sich aus dem Vergleich der beiden relationalen Beschreibungen D_A und D_B ergibt. Der allgemeine Abstand von D_A und D_B resultiert dann als der minimale Fehler E(f) für alle Abbildungen f. Um alle E(f) berechnen zu können, schlägt Shapiro /SHP 84/ ein Suchverfahren vor, das Tiefe zuerst vorgeht.

Das vorgestellte BMS soll als Modell eine umfassende Sicht des Problembereichs enthalten können. Deshalb erscheint es zu aufwendig, irgendeine Art von vollständiger Abbildung Bild - Modell anzustreben. Vielmehr sollte es möglich sein, einzelne Objektmodelle mit dem Bild zu vergleichen, ohne das gesamte Modell in die Betrachtung miteinzubeziehen. Aus diesem Grund wählten wir einen etwas anderen Ansatz für den Interpretationsprozeß, der mehr Gewicht auf lokale Auswertungen legt und von anderen Arbeiten am Lehrstuhl /SAG 85, HOF 82, GRE 84/ inspiriert ist. Im ersten Abschnitt dieses Kapitels wird die Bewertungsfunktion zum Vergleich von Komponenten aus Bild und Modell vorgestellt, während im zweiten Abschnitt die Vorgehensweise, der Kontrollalgorithmus, erklärt wird.

7.1 Bewertungsfunktion

Als lokale Bewertungsfunktionen für einzelne Parameter treten im Modell die Bedingungen, die einen Wert$\varepsilon[0,1]$ liefern, die Auswertung der maximalen und minimalen Anzahl gleicher Teile, die 0 oder 1 als Ergebnis hat, und Fuzzy-Funktionen auf. Die Auswertung der Fuzzy-Funktionen ergibt einen Wert $\varepsilon[0,1]$. Dabei steht z.B. im Modell für die Deckfläche eines Schlotes die in Bild 7.2 dargestellte Fuzzy-Funkton für den Radius. Eine aus dem Bild detektierte Ellipse oder ein Kreis mit dem Radius 0.3 m, 1.3m oder 3 m kann damit mit Sicherheit 0.2, 1 oder 0 als Deckfläche zugeordnet werden. Dabei wird natürlich der Radius alleine nicht signifikant sein. Mit den verschiedenen lokalen Bewertungsfunktionen muß nun eine globale Bewertung für die Zuordnung der Strukturen im Bild zum Objektmodell oder der Szene erzeugt werden.

Welche lokalen Bewertungen zusammengefaßt werden müssen, um eine gewünschte globale Bewertung zu erhalten, ist aus der Tabelle Modellbildung (Bewertung) in Abschnitt 5.6 ersichtlich. Für die Analyse ist dort die notwendige Information

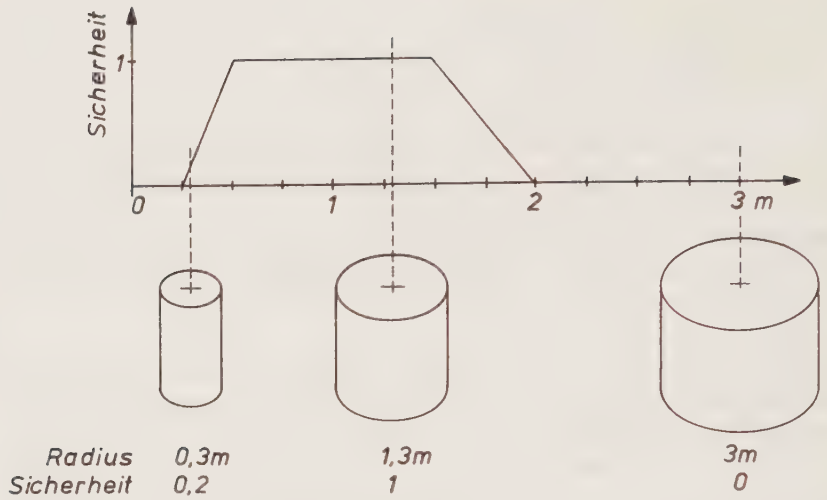


Bild 7.2: Auswertung einer Fuzzy-Funktion

zur Steuerung der Reihenfolge bei der Zusammenfassung von Bewertungen niedergelegt. Wie diese Tabelle verwendet werden kann, soll am Beispiel des Stuhls aus Bild 5.10 erläutert werden. Dazu wird im Modell nachgeschaut, welche Größen jeweils explizit für die entsprechenden Variablen eingesetzt werden. Da der Stuhl als Objekt modelliert ist, beginnt die Auswertung mit SICH(OBJEKT).

SICH(STUHL) = SICH(SITZ) § SICH(LEHNE) § SICH(ZUSHLEHNE) § SICH(BEIN1) § SICH(ZUSHBEIN1) § SICH(BEIN2) § SICH(ZUSHBEIN2) § SICH(BEIN3) § SICH(ZUSHBEIN3) § SICH(BEIN4) § SICH(ZUSHBEIN4) § (SICH(BED))*

In diesem Beispiel wird eine Aufspaltung der Bedingungen BED ausgelassen und auch nur ein Objektteil (LEHNE) weiterverfolgt. Dabei ergibt sich die Bewertung der Transformation des KOS der Lehne in das Objekt-KOS aus der entsprechenden Translation und Rotation in SICH(ZUSHLEHENE). Für die Lehne gilt weiter:

SICH(LEHNE) = SICH(VZLEHNE) § SICH(INTENSITÄT)

SICH(VZLEHNE) = SICH(GRUNDFLEHNE) § SICH(DECKFLEHNE) § SICH(HOEHE) § (SICH(BED))*

SICH(GRUNDFLEHNE) = SICH(VIERECKGL) § (SICH(BED))*

SICH(VIERECKGL) = SICH(KOX1) § SICH(KOY1) § SICH(KOX2) § SICH(KOY2) § SICH(KOX3) § SICH(KOY3) § SICH(KOX4) § SICH(KOY4)

Damit kann die Grundfläche bewertet werden. Nach entsprechender Bewertung der Deckfläche kann eine Bewertung für die Lehne errechnet werden usw. Für eine Analyse ist es aber auch möglich, zu einer gefundenen Fläche im Bild alle passenden Oberflächen des Modells zu berechnen und damit auf vorkommende

Primelemente zu schließen. Dies erleichtert der einfache Zugriff innerhalb der relationalen Datenbank auf alle Grund- und Deckflächen.

Es bleibt jetzt noch zu klären, wie die einzelnen lokalen Bewertungen zusammengefäßt werden, also welches Aussehen der Operator § hat. Dies entspricht dem letzten Schritt in den "test score semantics" nach Zadeh /ZAD 85/. Gegeben seien zwei Werte x_1 und x_2 aus zwei Einzeltests, § ließe sich als Vereinigung v dieser beiden Einzeltests ansehen. Dafür gibt Zadeh /ZAD 75/ die Minimumsbildung $MIN(x_1,x_2)$ an. Als weitere Möglichkeit wird eine starke Vereinigung $x_1 \underline{AND} x_2$ in Betracht gezogen, die als Multiplikation realisiert wird. Diese beiden Operatoren haben die unangenehme Eigenschaft, daß schon ein Einzeltest mit Bewertung 0 das Gesamtergebnis auf 0 drückt. Da nun in der Bildverarbeitung durchaus einzelne lokale Fehler bei der Extraktion der Merkmale zu erwarten sind, haben wir zwei weitere Operatoren für die Zusammenfassung der Einzeltests betrachtet, ein erhöhtes Minimum $MIN2(x_1,x_2)$ und den Mittelwert $MIT(x_1,x_2)$.

$MIN2(x_1,x_2) = MIN(1, (MIN(x_1,x_2)+1/4MAX(x_1,x_2))$

$MIT(x_1,x_2) = 1/2(x_1+x_2)$

Da für x_1 und x_2 nur Werte im Bereich 0,1 auftreten, gelten folgende Ungleichungen:

AND MIN MIN2

AND MIN MIT

Mit den verschiedenen Operatoren haben wir Bewertungen für den Vergleich von VZ aus dem Modell und synthetisch erzeugten Bildzylindern getestet (siehe Anhang). Dabei erwiesen sich das erhöhte Minimum MIN2 und der Mittelwert MIT als vorteilhaft, da sie auch einzelne Fehler tolerieren. Das erhöhte Minimum MIN2 ist allerdings von der Reihenfolge der Auswertung abhängig. Deshalb findet zur Zeit der Mittelwert für die Zusammenfassung der Einzeltests als Operator § Verwendung.

7.2 Kontrollalgorithmus

Mit der im vorherigen Abschnitt definierten Bewertungsfunktion kann man nun die im Sinne dieser Bewertungsfunktion beste Interpretation berechnen, indem man alle möglichen Interpretationen durchtestet. Dies dauert aber zu lange. Deshalb versucht man durch den Einsatz von Heuristik den Aufwand zur Berechnung der optimalen oder wenigstens einer suboptimalen Lösung merklich zu verringern.

Eine gängige Methode, die Suche nach der besten Zuordnung zu verkürzen, stellt

der A^*-Algorithmus dar (siehe Tafel 7.1). Der A^*-Algorithmus findet eine optimale Lösung, wenn die Bewertungsfunktion $g^*(v_i)$ monoton ist und für $h^*(v_i)$ gilt, daß $h^*(v_i)$ eine untere Schranke für $h(v_i)$ ist. Ein Beweis dieser Aussage findet sich z.B. bei Nilsson /NIL 82/. Beim Erkennen von teilweise verdeckten Werkstücken setzt Grebner /GRE 85, OTT 86/ den A^*-Algorithmus ein. Dabei wird

A^*-Algorithmus:

Gegeben: Sei G ein Graph mit Knoten $v_i \in V$, einem Startknoten $s \in V$ und Zielknoten $t_i \in V$.
Sei $f(v_i)$ eine Bewertungsfunktion , die für jeden Knoten v_i die Kosten für den billigsten Weg vom Startknoten s zu irgendeinem Zielknoten t_i über v_i angibt. Diese wird zerlegt in $f(v_i)=g(v_i)+h(v_i)$ mit $g(v_i)$ als den minimalen Kosten vom Startknoten s zum Knoten v_i und $h(v_i)$ als den minimalen Kosten vom Knoten v_i zu einem Endknoten t_i.
Meist ist nur eine Schätzfunktion $f^*(v_i)=g^*(v_i)+h^*(v_i)$ verfügbar. Zusätzlich benötigt man eine OFFEN-Liste und eine GESCHLOSSEN-Liste.

Gesucht: Optimaler Weg bzg. Kosten vom Startknoten s zu einem Endknoten t_i.

1. Berechne $f^*(s)$ und speichere s in der OFFEN-Liste.
2. Falls OFFEN-Liste leer → Fehler.
3. Streiche bestbewerteten Knoten v_n in der OFFEN-Liste und speichere v_n in die GESCHLOSSEN-Liste.
4. Ist v_n ein Zielknoten, so ist die gesuchte Lösung gefunden. Verfolge die in Schritt 5 aufgestellten Zeiger zurück → fertig
5. Entwickle v_n, d.h. stelle alle Nachfolger v_{ni} fest und berechne $f^*(v_{ni})$. Ist v_{ni} bisher nicht aufgetreten, so speichere ihn in die OFFEN-Liste und setze einen Zeiger auf v_n. Ist v_{ni} bereits in der OFFEN oder GESCHLOSSEN-Liste mit einer besseren Bewertung, muß nichts weiter gemacht werden. Ist v_{ni} bereits in der OFFEN oder GESCHLOSSEN-Liste mit einer schlechteren Bewertung, so wird die neue Bewertung eingetragen und alle entsprechenden Zeiger müssen umgesetzt werden.
6. Gehe zu Schritt 2.

Tafel 7.1: A^*-Algorithmus

das zu erkennende Wekstück mit einigen charakteristischen Merkmalen modelliert, solche Merkmale werden dann aus dem Bild extrahiert. Indem jedem Merkmal aus dem Modell alle im Bild gefundenen Kandidaten zugeordnet werden, wird ein vollständiger Abarbeitungsgraph aufgebaut, indem dann mit dem A*-Algorithmus der optimale Pfad gesucht wird. Auch im nuklear-medizinischen Bildanalysesystem am hiesigen Institut wird der A*-Algorithmus zur Kontrolle der Analyse verwendet /HOF 82/. Der A*-Algorithmus steuert dabei die Suche durch das zugrundeliegende semantische Netz, indem er den optimalen Pfad zwischen einem angegebenen Zielkonzept (meist dem Konzept "Diagnose") und den von der Vorverarbeitung gelieferten Konturverläufen des Herzens berechnet.

In beiden Systemen wird der vollständige Ableitungsgraph modellgesteuert durchsucht. Eine Verbesserung erhofft man sich durch eine gemischte ("top-down, bottom-up") Kontrolle. Dies schlägt Eichhorn /EIC 86/ für die Suche in semantischen Netzen vor. Dabei werden in einem Vorlauf Produktionen erzeugt, die mittels Attributen aufgrund der aus dem Bild extrahierten Merkmale einige vorläufige Zielkonzepte auswählen. Diese werden dann modellgesteuert mit dem A*-Algorithmus expandiert und instantiiert. Mit diesen vorläufigen Zielkonzepten werden nun neue, in der Hierarchie des Netzes höher stehende Konzepte als Zielkonzepte ausgewählt. Nach diesem "bottom-up"-Schritt werden die neuen Zielkonzepte wieder expandiert und instantiiert. Auf dies Weise kann die Kontrolle noch mehrfach hin- und herspringen, bis ein endgültiges Zielkonzept gefunden ist. Ebenfalls bidirektional ist der Kontrollalgorithmus, den Schärf /SHF 85/ vorstellt, allerdings sehr viel einfacher und deshalb auch beschränkter in den Anwendungsmöglichkeiten. Es geht um die Festlegung der dreidimensionalen Lage eines Objekts in einem Bild anhand eines Modells des Objekts (hier: ein VW-Bus). Dabei wird komponentenweise (Regionen, Kanten usw.) eine Vorklassifikation vorgenommen. Da zur Festlegung der dreidimensionalen Lage drei Komponenten benötigt werden, werden die beiden bestbewerteten zu einem Paar zusammengefaßt und mithilfe des Modells die Lage einer dritten Komponente (Ellipsenbahnen) hypothetisiert, wobei mögliche Prototypen projiziert werden. Hier erfolgt nun ein Vergleich mit bereits aus dem Bild gewonnenen Komponenten oder eine modellgesteuerte Nachsegmentation.

Ähnlich den beiden letzten Verfahren ist die von uns verwendete Kontrolle auch bidirektional /KOR 86, HES 86b/. Dabei ist die Vorgehensweise objektorientiert. Zuerst werden datengesteuert Hinweise auf Objekte gesucht. Aus diesen Hinweisen wird ein Objekt ausgewählt, das dann modellgesteuert mit dem A*-Algorithmus interpretiert wird. Ist die Interpretation eines Objekts abgeschlossen, wird

ein Hinweis auf das nächste Objekt wiederum datengesteuert gesucht, modellgesteuert interpretiert usw., bis alle Objekte abgearbeitet sind. In einer zweiten Phase werden dann die Regionen des Bildes auch mit Hilfe des A^*-Algorithmus interpretiert. Der gesamte Kontrollalgorithmus hat dann folgendes Aussehen /KOR 86/:

Gegeben: Modell und Bild (siehe Abschnitt 8.1)
Gesucht: Interpretation der im Bild von der Vorverarbeitung gefundenen Strukturen (z.Zt. verallgemeinerte Zylinder und Regionen) anhand des Modells

Phase 1: Interpretation von Objekten

1. Nimm beliebigen, noch nicht interpretierten Bildzylinder BZ_k, vergleiche BZ_k mit allen Modellzylindern MZ_i mittels der Bewertungsfunktion aus Abschnitt 7.1.
2. Wähle den Modellzylinder MZ_k mit der besten Bewertung aus und suche im Modell den Modellzylinder MZ_n, der das Hauptteil des Objekts ist, zu dem auch MZ_k gehört. Speichere alle zu diesem Objekt gehörigen Modellzylinder MZ_i in die Liste TEILE ab.
3. Suche zu dem Hauptteil MZ_n den am besten passenden Bildzylinder BZ_n.
4. Suche im Bild alle Bildzylinder BZ_i, die mit BZ_n in örtlichem Zusammenhang stehen und speichere sie in die Liste PART.
5. Interpretiere die Bildzylinder aus der Liste PART durch die Modellzylinder aus der Liste TEILE mit Hilfe des A^*-Algorithmus.
6. Versehe die Bildzylinder BZ_j entlang des optimalen Pfades, der in Schritt 5 berechnet wurde, mit der entsprechenden Interpretation, entferne sie aus der Liste der noch nicht interpretierten Bildzylinder und lösche die Inhalte der Listen TEILE und PART.
7. Sind alle Bildzylinder interpretiert, setze mit Phase 2 fort, sonst gehe zu Schritt 1.

Phase 2: Interpretation der Regionen

8. Spanne mit allen Regionen im Bild und allen Regionen im Modell einen Graphen auf und suche mittels des A^*-Algorithmus und der Bewertungsfunktion aus Abschnitt 7.1 den optimalen Pfad. Damit sind alle Regionen im Bild interpretiert.
9. Für alle Flächen des Bildes, die sich jeweils aus mehreren Regionen zusammensetzen, treffe eine Mehrheitsentscheidung mit den Interpretationen der dazugehörigen Regionen.

Testläufe mit diesem Kontrollalgorithmus haben selbst bei kleinem Modell und kleinem Bild eine Zeitverbesserung um den Faktor 5 gezeigt. Zur Verbesserung der Erkennungssicherheit trägt bei, daß ähnliche Teile unterschiedlicher Objekte nicht gemischt werden, da sie während der Hypothetisierung in Schritt 2 nicht gemeinsam auftreten können. Die genauen Ergebnisse mit verschiedenen Beispielen werden im nächsten Kapitel erläutert.

8. Zwei Beispiele für die Modellbildung

In Kapitel 5 wurde ein Bildanalysemodellierungssystem vorgestellt, das Strukturen für die Modellbildung und die Darstellung der Modelle beinhaltet. In diesem Kapitel soll anhand von zwei Beispielen gezeigt werden, wie Modelle in diesem Formalismus dann aussehen. Dabei muß die Modellerstellung (Abschnitt 8.1) interaktiv erfolgen. Die Analyse von interaktiv segmentierten Bildern (Abschnitt 8.2) aus den Problembereichen natürliche Szenen (Abschnitt 8.3) und industrielle Bilder (Abschnitt 8.4) wird mit dem im letzten Kapitel erläuterten Kontrollalgorithmus durchgeführt.

8.1 Modellerstellung

Bevor das Modell in den Rechner eingegeben werden kann, muß ein Operator mit eigener Intuition den Problembereich durchdringen und für die Modellerstellung geeignet aufbereiten. Dazu müssen u.a. folgende Fragen geklärt werden:

- Welche Objekte sollen modelliert werden und in welcher örtlichen Beziehung stehen sie zueinander?
- Wie sind diese Objekte aufgebaut?
- Welche Teile enthalten diese Objekte und in welcher örtlichen Beziehung stehen diese Teile zueinander?
- Wie detailliert sollen die Objekte modelliert werden?
- Mit welchen verallgemeinerten Zylindern werden die Objektteile modelliert?
- Welche Oberflächenattribute sollen an welche Oberflächen angehängt werden?
- Welche Variationen sollen für die Objektmodelle zugelassen werden?
- Wie werden diese Variationen erreicht?
- Wie sehen die nötigen Fuzzy-Funktionen aus?
- Welche Objekte werden zu Objektgruppen zusammengefaßt und in welcher örtlichen Beziehung stehen die Objekte zueinander?
- Welche Regionen sollen modelliert werden?

Insgesamt kann ein Modell dann in folgenden Schritten erstellt werden:

1. Zusammenstellung der Objekte des Problemkreises, die modelliert werden sollen und Erfassung ihrer örtlichen Beziehungen zueinander
2. Aufstellung eines Objektmodells mit den entsprechenden Teilen, Oberflächenattributen, Fuzzy-Funktionen und Bedingungen
3. Eintrag in die Datenbank
4. Visualisierung von extremen Ausprägungen des Objektmodells zur Prüfung der semantischen Konsistenz

5. Falls alle Objektmodelle erstellt, weiter mit Schritt 6, sonst Erstellung des nächsten Objektmodells (Schritt 2)
6. Definition von Objektgruppen und Eintrag in die Datenbank mit Visualisierung
7. Modellierung von Regionen und Eintrag in die Datenbank

Zur Unterstützung desjenigen, der die Modelle erstellt, stehen Programme zum Eintrag in die Datenbank und zur Visualisierung der Objektmodelle zur Verfügung. Der Datenbankeintrag /LAU 86/ wird über eine komfortable Benutzerschnittstelle abgewickelt, die Masken erzeugt (siehe Bild 8.1), die nur noch gefüllt werden müssen. Das Programm überprüft Primärschlüsselattribute auf Eindeutigkeit, die Einhaltung von Attributformaten und Wertebereichseinschränkungen und ermöglicht Änderungen und die Verbesserung von Fehlern. Zusätzlich führt es den Operator durch das Modell, indem es die auszufüllenden Relationen in einer sinnvollen Reihenfolge zur Eintragung vorschlägt.

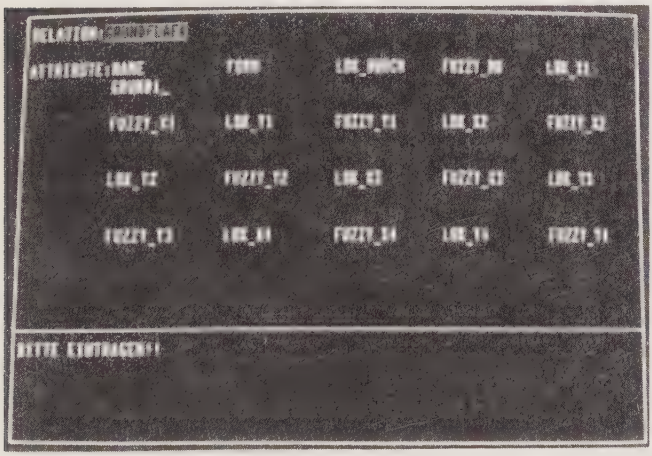


Bild 8.1: Eingabemaske zum Eintrag in die Datenbank

Für die Visualisierung existiert ein Programm, das auf der Basis des Grafikpakets GRIBS Objekte oder ganze Szenen auf einem Grafikbildschirm darstellt. Da allerdings keine weiteren Hilfsmittel zur Verfügung stehen, beschränken sich die Ausgabemöglichkeiten auf Drahtgitterdarstellungen der Objekte. Während des Programmlaufs können dabei die durch Fuzzy-Funktionen definierten Parameter des Objektmodells interaktiv fixiert werden, so daß verschiedene extreme Ausprä-

gungen des Objektmodells erzeugt werden können. Neben dem Grafikbildschirm ist auch eine Ausgabe auf einem grafikfähigen Matrixdrucker möglich. Verschiedene Bilder aus den Kapiteln 5 und 8 sind auf diesem Weg entstanden.

Die übrigen Schritte bei der Modellerstellung, also vor allem auch die Klärung der eingangs aufgeworfenen Fragen, bleiben dem Operator überlassen. Allererste Versuche, die Zerlegung von Objektmodellen zu automatisieren, finden sich in /SIE 86/. In wieweit sich der gesamte Komplex der Modellerstellung automatisieren läßt, wird in späteren Arbeiten geklärt werden müssen.

8.2 Erzeugung von Testbildern

Für den Test des Kontrollalgorithmus aus Kapitel 7 benötigt man Aufnahmen, aus denen Strukturen zum Vergleich mit dem Modell extrahiert sind. Dies wird in einem Bildanalysesystem von der Bildvorverarbeitung und Bildverarbeitungsalgorithmen übernommen. Die gefundenen Strukturen (hier: verallgemeinerte Zylinder und Regionen) werden in einer relationalen Datenbank abgelegt, die solche Relationen enthält, die den Relationen des Modells ähnlich sind (siehe Bild 8.2 und Anhang). Es fehlt natürlich dabei die Klassenbildung. In den Relationen VERZYL und GRUNDF ist die Geometrie der verallgemeinerten Zylinder abgelegt, VERZYLAT enthält die Oberflächenattributen und FLAECHEN sowie REGIO die Regionen. Alle lokalen Bezugspunkte sind entweder dreidimensonal im Szenenkoordinatensystem errechnet oder sie beziehen sich einfach auf die Pixel des Bildes (bei Regionen). Für die 3D-Transformationen steht die Relation ZUSH zur Verfügung. Um als Attributwerte keine Mengen zulassen zu müssen (also um die 1. Normalform der Relationen zu erhalten), sind die Regionen des Bildes in Dreiecke und/oder Vierecke aufgeteilt. Diese Dreiecke bzw. Vierecke sind mit den Oberflächenattributen in der Relation REGIO abgelegt, während die Relation FLAECHEN die Teile einer Region im Bild zusammenfaßt.

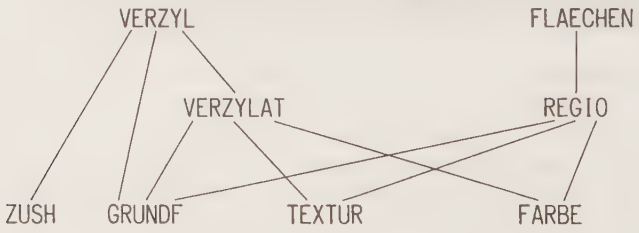


Bild 8.2: Relationen des Bildes

Da in dem vorhandenen Bildanalysesystem keine Bildvorverarbeitung existiert, muß der Operator diese simulieren und die entsprechenden Relationen von Hand füllen. Dies ist auf zwei Arten möglich, entweder er erzeugt ein synthetisches Bild (siehe Bild 8.6 und 8.7) oder er spielt für eine Photografie Bildvorverarbeitung ein (siehe Bild 8.8 - 8.10). Eine der wichtigsten Weiterentwicklungen wird der Einbau eines Bildverarbeitungsprogramms in das Bildanalysesystem sein.

8.3 Natürliche Szenen

Unter natürlichen Szenen wollen wir solche Bilder verstehen, die aus der Erfahrungswelt des Menschen stammen und bei denen während der Aufnahme wenig Einfluß auf die Umgebungsbedingungen genommen werden kann. Mit dieser Definition fallen darunter das Innere einer Fabrikhalle ebenso wie eine Büroszene. Auch die berühmten Vorstadtszenen (vgl. /HAN 78/, /LEV 78/ oder /OHT 84/) zählen dazu. Wir betrachten hier eine allgemeine Vorstadtszene und einen Blick aus dem Erlanger Informatikhochhaus. Um den Prozeß der Modellerstellung zu verdeutlichen, gehen wir die im Abschnitt 8.1 aufgestellten Schritte einmal durch.

8.3.1 Allgemeine Vorstadtszene

Das Modell soll hier aus einem Einfamilienhaus, einem Kirchenschiff, einem Kirchturm und einem Baum bestehen. Kirchenschiff und Kirchturm sollen in enger Nachbarschaft stehen. Für die örtlichen Beziehungen der beiden anderen Objekte zueinander und zur Kirche gibt es keine Aussage. Beginnen wir mit der Modellierung des Einfamilienhauses (vgl. Schritt 2 aus Abschnitt 8.1). Es ist aufgebaut aus dem Haus, einem Dach, zwischen 3 und 8 Fenstern und einer oder zwei Türen. Diese Teile werden als VZ modelliert, wobei alle Teile aus quaderförmigen Zylindern bestehen. Grund- und Deckfläche sowie die Höhe werden mittels Fuzzy-Funktionen variabel definiert. Bei dem Hauskörper bedeutet dies, daß sowohl für die Höhe als auch für die Koordinaten der Grund- und Deckfläche Fuzzy-Funktionen erzeugt werden müssen. Dies muß der Operator erledigen. Ausgehend von der Fuzzy-Funktion F(u;A,B,C,D), die durch die vier Parameter A, B, C und D festgelegt ist (siehe Abschnitt 5.42), gelten für den Hauskörper die in Tabelle 8.1 aufgezeichneten Werte. Dabei ist die Deckfläche mit der Grundfläche identisch. Die Modellierung der Oberflächenattribute erfolgt in der Relation OBJTEILE_K (siehe Abschnitt 5.7.2), wobei dort für jede Oberfläche entsprechend der Numerierung aus Bild 5.11 die Intensität und Verweise auf andere Oberflächenattribute niedergelegt werden. Die Definition der zulässigen Werte erfolgt wieder mittels Fuzzy-Funktionen. So besitzt der Hauskörper z.B. für die

		A	B	C	D
HÖHE		2,0	2,5	4,0	5,0
GRUNDFLÄCHE	x_1	-1,0	0,0	0,0	1,0
	y_1	-1,0	0,0	0,0	1,0
	x_2	-1,0	0,0	0,0	1,0
	y_2	3,0	4,0	10,0	15,0
	x_3	3,0	5,0	15,0	20,0
	y_3	3,0	4,0	10,0	15,0
	x_4	3,0	5,0	15,0	20,0
	y_4	-1,0	0,0	0,0	1,0

Tabelle 8.1: Fuzzy-Funktionen für den Hauskörper

Intensität der Oberfläche 2 die durch die vier Parameter A = 10.0, B = 10.0, C = 40.0 und D = 60.0 beschriebene Fuzzy-Funktion. In ähnlicher Weise werden das

		A	B	C	D
TRANSLATION	x	0,3	0,4	1,0	1,5
	y	-0,1	0,0	0,0	0,1
	z	0,3	0,4	1,0	1,5
ROTATION	x	-2,0	-1,0	1,0	2,0
	y	-2,0	-1,0	1,0	2,0
	z	-2,0	-1,0	1,0	2,0

Tabelle 8.2: Fuzzy-Funktionen für die Transformation des KOS eines Fensters in das Objekt-KOS

Dach, bei dessen viereckiger Deckfläche je 2 Koordinaten zusammenfallen, die Fenster und die Türen modelliert. Fenster und Türen stellen dabei sehr schmale quaderförmige VZ dar. Nun muß die örtliche Lage der einzelnen Objektteile innerhalb des Objekts bestimmt werden. Der Hauskörper trägt das Objekt-KOS, und es muß nun in der Relation TEILE_K für jedes Objektteil die Transformation des lokalen KOS in das Objekt-KOS mittels Fuzzy-Funktionen angegeben werden, ausgedrückt in Translation und Rotation in x-, y- und z-Richtung. Diese Parameter werden in ZUSHANG_K abgelegt. Tabelle 8.2 enthält die Transformation für die Lage eines Fensters innerhalb des Hauskörpers. Um das Modell des Einfamilienhauses zu vollenden, sind nun noch einige Bedingungen notwendig, um zwar physikalisch mögliche, aber semantisch nicht zulässige Objekte auszuschließen. Dazu gehört die Bedingung, daß Grund- und Deckfläche rechteckig und gleich groß sind, ebenso wie die Bedingung, daß die Grundfläche des Daches der Deckfläche des Hauskörpers entsprechen muß. Ist diese Bedingung nicht erfüllt, so erhält man das Objekt, das in Bild 5.18 gezeigt wird. Mit der Definition von einigen weiteren Bedingungen ist Schritt 2 für die Objektklasse Einfamilienhaus abgeschlossen.

Der Eintrag in die relationale Datenbank (Schritt 3) erfolgt mit dem oben (Abschnitt 8.1) beschriebenen Modelleingabewerkzeug. Dabei sind Tupel in die Relationen EINZELOBJ_K, OBJTEILE_K, TEILE_K, VERALLZYLK, GRUNDFLAEK, ZUSHANG_K, FUZZY, GRUNDFLAEB, OBJBED und falls erforderlich in OBERFARBEK sowie in TEXTUR_K einzufügen.

Im 4. Schritt kann nun eine Visualisierung von Instanzen des Objektmodells durchgeführt werden, damit man überprüfen kann, ob das Objektmodell nicht semantisch unzulässige Instanzen zuläßt. Hierzu werden in einem interaktiven Programm Werte für die verschiedenen Fuzzy-Funktionen angegeben, die jeweils innerhalb des Bereichs liegen sollen, der die Bewertung 1 ausdrückt. Zwei extreme Ausprägungen des Objektmodells Einfamilienhaus, allerdings ohne Oberflächenattribute, zeigt Bild 8.3. Die beiden Instanzen variieren sowohl die Ausmaße als auch die Anzahl der Teile und deren örtliche Beziehungen innerhalb des Objektmodells.

Die Schritte 2 bis 4 werden für die anderen Objekte der Szene wiederholt. Das Kirchenschiff besteht ähnlich dem Haus aus dem Körper, dem Dach, vier bis fünfzehn Fenstern und einem bis drei Portalen. Alle Objektteile werden durch quaderförmige VZ beschrieben und mit ihren Oberflächenattributen sowie den notwendigen Bedingungen in die Datenbank eingetragen. In gleicher Weise ver-

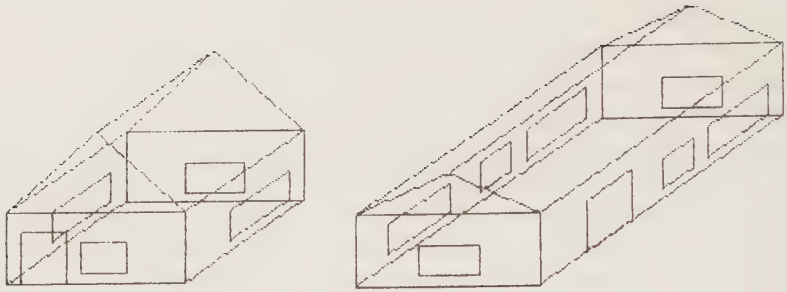


Bild 8.3: Zwei Ausprägungen des Objektmodells Einfamilienhaus

fährt man mit dem Kirchturm. Zylinderförmige Primelemente werden dagegen fur die Objektteile des Baums, Stamm und Krone, verwendet. Die Approximation der Krone durch einen Zylinder erscheint für viele Bildanalysezwecke geeignet, da eine Modellierung der einzelnen Äste und Blätter sowohl bei der Erstellung als auch bei der Nutzung sehr aufwendig ist. Gemäß der Syntax benötigen wir lediglich die Angabe des Radius von Grund- und Deckfläche sowie der Höhe, ausgedrückt jeweils in Fuzzy-Funktionen, um die Geometrie von Stamm und Krone (siehe Tabelle 8.3) zu definieren. Zusammen mit den Oberflächenattributen und den Bedingungen können diese Tupel in die Datenbank eingetragen werden. Mit einer möglichen Visualisierung sind die Schritte 2 bis 4 abgeschlossen.

	A	B	C	D
HÖHE	5,0	6,0	11,0	15,0
RADIUS GRUNDFLÄCHE	1,5	2,0	4,0	5,0
RADIUS DECKFLÄCHE	0,3	0,4	1,5	2,0

Tabelle 8.3: Fuzzy-Funktionen für die Krone des Baumes

Die Zusammenfassung von Objektmodellen zu Objektgruppen (Schritt 6) benötigt die Festlegung eines Hauptobjekts, das das Objektgruppen-KOS trägt. Bei der allgemeinen Vorstadtszene gibt es zwei Objektgruppen. Eine davon besteht aus

drei Einfamilienhäusern, die in einer Reihe stehen und von denen ein willkürlich ausgewähltes mit dem Objektgruppen-KOS assoziiert wird. Die zweite Objektgruppe heißt Kirche und beinhaltet Kirchenschiff und Kirchturm, wobei als örtliche Beziehung sowohl ein Aneinanderstoßen der beiden Objekte als auch ein gewisser Abstand zwischen beiden (siehe Bild 8.4) in die Transformation hineinmodelliert ist. Abgelegt werden die Objektgruppen in die Relationen OBJGRUPPEK, ZUSHANG_K und FUZZY.

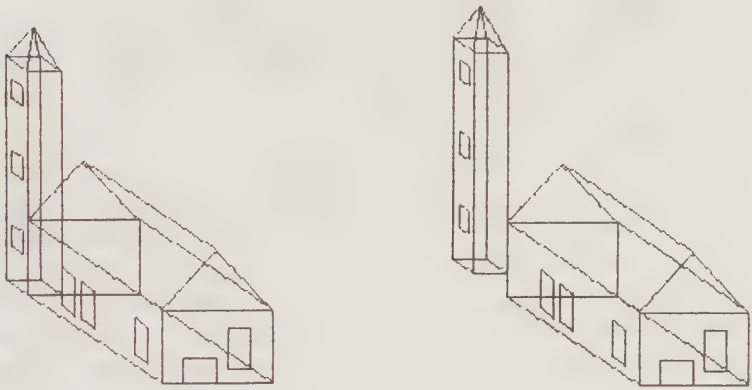


Bild 8.4: Zwei Ausprägungen der Objektgruppe Kirche

Zuletzt werden noch Regionen modelliert. Darunter fallen alle Objekte, die nicht oder nur schwer in VZ zu fassen sind. Dies sind bei dieser Vorstadtszene Himmel, Wald, Weg und Gras. Allerdings stehen zu Beschreibung und später zur Analyse lediglich die Oberflächenattribute zur Verfügung. Da in diesem Modell sogar nur die Intensität für die Regionen ausgefüllt ist, ist eine unzweideutige Interpretation der Regionen nicht immer möglich. Mit Oberflächenattributen wie Farbe oder Textur sollte dies besser werden.

Damit ist ein Modell für eine einfache Szene erstellt. Die Objekte sind als Objektklasse modelliert, so daß sich von jedem Objektmodell beliebig viele Ausprägungen erzeugen lassen. Dies kann man interaktiv mit einem Programm, das aus dem Modell ein Bild erstellt, tun. Bild 8.5 ist ein Beispiel hierfür. Allerdings findet keine Überprüfung von stabilen Lagen und anderen physikalischen Gesetzen statt, wie es in dem System von Adorni et al. /ADO 84/ angestrebt wird.

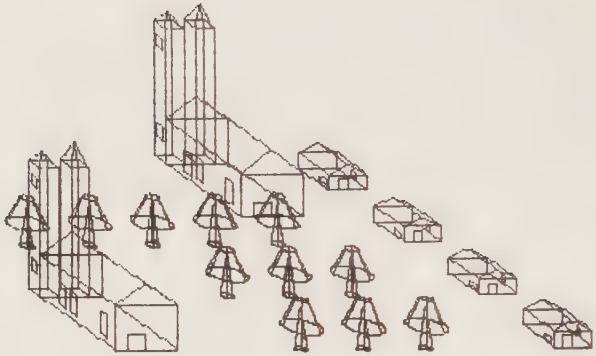


Bild 8.5: Aus dem Modell abgeleitetes Bild

Diese allgemeine Vorstadtszene wurde mit zwei Bildern getestet. Die beiden Testbilder HAUS (siehe Bild 8.6) und KIRCHE (siehe Bild 8.7) wurden synthetisch erstellt und anhand des Modells interpretiert. Dazu wurde der in Kapitel 7 beschriebene Kontrollalgorithmus verwendet. Das Testbild HAUS besteht aus 7 verallgemeinerten Zylindern, die in der Relation VERZYL niedergelegt sind und deren dreidimensionale Lage in der Relation ZUSH festgehalten ist. Die Be-

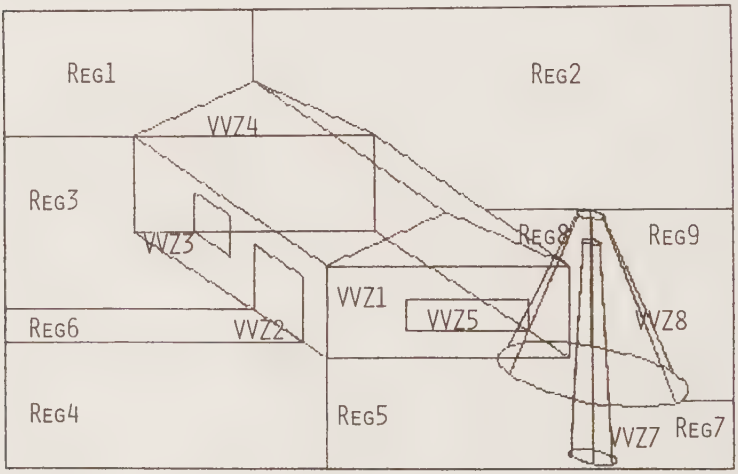


Bild 8.6: Testbild HAUS

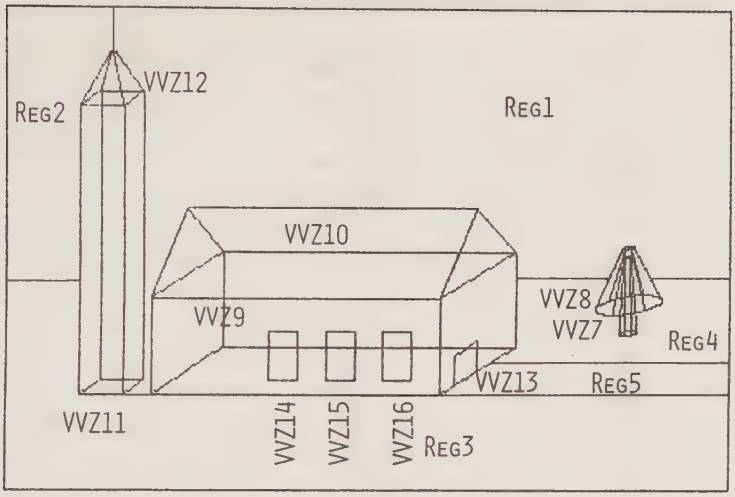


Bild 8.7: Testbild KIRCHE

schreibung erfolgt mit Grundfläche (Relation GRUNDF), Deckfläche (Relation GRUNDF), Höhe und Oberflächenattributen (Relation VERZYLAT), von denen allerdings nur die Intensität im Bild auftritt. Daneben sind noch 2 Flächen im Testbild HAUS, die sich in 9 Regionen (Relation REGIO) aufspalten. Diese Regionen werden als viereckige Flächen (Relation GRUNDF) und mit Oberflächenattributen (Relation REGIO) im Bild erkannt bzw. synthetisiert. Das Testbild KIRCHE besteht aus 10 verallgemeinerten Zylindern und 5 Regionen. Bei der Interpretation werden nach Möglichkeit allen verallgemeinerten Zylindern und allen Regionen im Bild Namen zugewiesen. Die Ergebnisse für beide Testbilder sind in den Tabellen 8.4 (Testbild HAUS) und 8.5 (Testbild KIRCHE) aufgeführt.

Auffallend beim Testbild HAUS sind die Fehler bei der Interpretation von VVZ2 und VVZ3. Ein Blick auf die Ergebnisse der Bewertungsfunktion (siehe Anhang) zeigt, daß für den Bildzylinder VVZ2 sogar ein Kirchenfenster lokal die beste Bewertung erbracht hätte. Da aber der Kontrollalgorithmus im Kontext von Objektmodellen vorgeht, scheidet diese Interpretation aus und die zweitbeste lokale Bewertung tritt in Kraft, die auch noch falsch ist. Eine Beseitigung dieses Fehlers kann nur durch zusätzliche Information wie dem lokalen Zusammenhang oder zusätzliche Bedingungen wie "ein Fenster darf nicht zum Boden reichen" erzielt werden. Für den Bildzylinder VVZ5 allerdings genügt das einge-

1.) verallg. Zylinder im Bild:

Primkoerper	Interpretation
VVZ1	HAUS
VVZ2	FENSTER
VVZ3	TUER
VVZ4	DACH
VVZ5	FENSTER
VVZ7	STAMM
VVZ8	KRONE

2.) Regionen des Bildes:

Bildregionen	Interpretation
REG9	HIMMEL
REG8	HIMMEL
REG7	HIMMEL
REG6	WEG
REG5	HIMMEL
REG4	HIMMEL
REG3	HIMMEL
REG2	HIMMEL
REG1	HIMMEL

Tabelle 8.4: Interpretationsergebnisse für das Testbild HAUS

1.) verallg. Zylinder im Bild:

Primkoerper	Interpretation
VVZ9	KISCHIFF
VVZ10	KIDACH
VVZ13	KIPORTAL
VVZ14	KIFENST
VVZ15	KIFENST
VVZ16	KIFENST
VVZ11	KITURM
VVZ12	KITUDACH
VVZ7	STAMM
VVZ8	KRONE

2.) Regionen des Bildes:

Bildregionen	Interpretation
REG5	WEG
REG4	HIMMEL
REG3	HIMMEL
REG2	HIMMEL
REG1	HIMMEL

Tabelle 8.5: Interpretationsergebnisse für das Testbild KIRCHE

setzte Wissen, um eine Fehlinterpretation zu vermeiden. Die alleinige Anwendung der Bewertungsfunktion errechnet als beste Interpretation ein Kirchenfenster. Somit erbringt der Einsatz des Modells eine Verbesserung der Analyseergebnisse. Durch noch mehr Einbringen von Modellwissen in den Analyselauf können noch weitere Fehler vermieden werden. Die gleiche Aussage läßt sich auch für die Interpretation der Regionen machen. Da derzeit nur die Intensitätswerte verglichen werden, kann bei der Interpretation keine Unterscheidung zwischen Himmel und Gras gefällt werden. Deswegen werden die entsprechenden Regionen fehlinterpretiert. Auch hier muß zusätzliches Wissen über Oberflächenattribute (Farbe, Textur o.ä.) oder die örtlichen Beziehungen im Bild ("Himmel ist oben, Gras ist unten") für eine Verbesserung der Interpretation herangezogen werden.

Beim Testbild KIRCHE ist die Interpretation vollständig richtig. Allerdings würde sich auch dort eine Fehlinterpretation ergeben, wenn der Kirchturm direkt an der Kirche stehen würde. Der Grund liegt auch hier in der unvollständigen Auswertung des Modellwissens durch den Kontrollalgorithmus. Die zusätzliche Berücksichtigung von lokalen Beziehungen von Objekten und Objektteilen, die im Modell niedergelegt sind, würde diesen Fehler beheben. Für die Regionen gilt hier das gleiche wie für das Testbild HAUS. Die Unterscheidung von Gras und Himmel ist allein auf Intensität basierend nicht möglich, und dies führt zu einigen Fehlinterpretationen.

Die Laufzeiten dieses Kontrollalgorithmus sind ziemlich hoch. Bei beiden Testbildern liegt die Laufzeit bei je über einer Stunde. Ein Großteil der Zeit wird dabei in Datenbankabfragen verbracht, da ja Bild und Modell in einer relationalen Datenbank abgelegt sind. Solange Bild und Modell nicht umfangreicher sind, ist der Überhang an Verwaltung der wegen des Datenbankformalismus zu bearbeiten ist, sicher nicht notwendig. Es wird aber dadurch gezeigt, in welcher Allgemeingültigkeit mit dem vorgestellten BMS Bilder analysiert werden können. Dies war das vordringliche Anliegen beim Entwurf des Kontrollalgorithmus, der immerhin gegenüber einer vollständigen modellgesteuerten Suche mit dem A^*-Algorithmus bei der Laufzeit den Faktor 6 einspart /KOR 86/.

8.3.2 Ein Blick aus dem Fenster

Als etwas umfangreichere Szene wurde der Blick aus einem Fenster des Erlanger Informatikhochhauses modelliert. Verfolgen wir zur Modellerstellung nochmals die Schritte aus Abschnitt 8.1. Wir nehmen in das Modell drei Objekte auf (Schritt 1), ein Bürogebäude, eine Werkstatt und einen Schlot. Dabei gibt es

keine örtlichen Beziehungen der drei Objekte zueinander, die im Modell niedergelegt werden.

Das Gebäude besteht aus einem Körper, 20 bis 60 Fenstern, 2 oder 3 Dachaufbauten und 1 oder 2 Türen (Schritt 2). Alle Objektteile sind als quaderförmige Zylinder definiert, wobei allen Fenstern derselbe variabel gehaltene VZ zugrundeliegt. Definiert werden die VZ wieder mit Grundfläche, Deckfläche, Höhe, Oberflächenattributen und Bedingungen. Bei der Angabe, wo die Fenster im Gebäude liegen, muß darauf geachtet werden, daß diese Transformationen in geeigneter Weise festgelegt werden. Es muß sowohl bei 20 als auch bei 60 Fenstern möglich sein, die Fenster gleichmäßig über das Gebäude zu verteilen, da dies in der Realität meist der Fall ist. Entsprechend müssen die Fuzzy-Funktionen, die Translation und Rotation in der Relation ZUSHANG_K beschreiben, diese Variabilität zulassen. Sind alle für die Modellierung des Gebäudes notwendigen Größen zusammengetragen, so erfolgt der Eintrag in die Datenbank (Schritt 3) und, falls erwünscht, eine Visualisierung verschiedener Instanzen des Objektmodells (Schritt 4). Ebenso muß noch für die beiden anderen Objekte verfahren werden. Die Werkstatt gliedert sich in ein Gebäude, 7 bis 10 Dachrippen, und 1 bis 2 Türen, während der Schlot aus einem zylindrischen VZ besteht. Objektgruppen werden keine modelliert (Schritt 6), so daß nur noch verschiedene Regionen ins Modell aufgenommen werden müssen (Schritt 7). Hierfür greifen wir

Bild 8.8: Ein Blick aus dem Fenster

auf die vier Regionen Himmel, Gras, Wald und Weg aus Abschnitt 8.3.1 zurück. Damit ist das Modell für diese Szene fertig. Die beiden Modelle aus dem vorherigen und aus diesem Abschnitt können auch in einer Datenbank vereinigt werden. Dabei muß nur auf die Eindeutigkeit der Namensgebung geachtet werden.

Als Grundlage für ein Testbild dient ein Schwarzweißfoto (siehe Bild 8.8), das von Hand segmentiert wurde (siehe Bild 8.9). Aus diesem segmentierten Bild wurden VZ und Regionen extrahiert (siehe Bild 8.10), die in die relationale Datenbank entsprechend Abschnitt 8.2 gespeichert wurden. Damit ist das Testbild Universität erstellt. Es enthält 44 VZ und 2 Regionen.

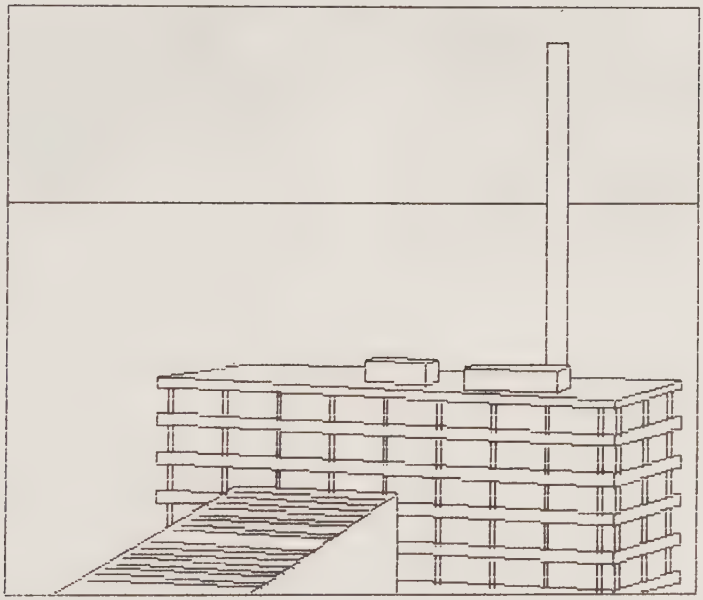


Bild 8.9: Handsegmentierung von Bild 8.8

8.4 Industrielle Szenen

Bei den industriellen Szenen geht es um Fragestellungen, die sich mit der Fertigung von Produkten in einer Fabrik beschäftigen. Eine Aufgabenstellung daraus, die Qualitätskontrolle, haben wir in Kapitel 1 angerissen. Daneben gibt es noch das Erkennen und Greifen von Werkstücken, die Montage von Teilen und

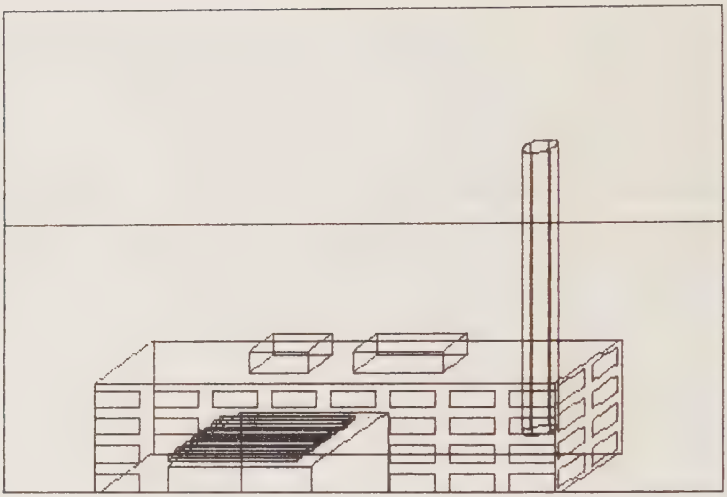


Bild 8.10: Testbild UNIVERSITÄT

die Überwachung von Fertigungszellen. Die letzten beiden Aufgabenstellungen sind für die industrielle Bildverarbeitung noch kaum bis gar nicht erschlossen. Neben den Entwurfskriterien, die im Forschungsbereich gelten, Durchführbarkeit und Fehlerhäufigkeit, treten im industriellen Bereich noch Kriterien wie Geschwindigkeit, Wirtschaftlichkeit, Zuverlässigkeit und Bedienbarkeit, die wir aber nicht weiter erörtern wollen.

Ein wesentlicher Unterschied zu den natürlichen Szenen besteht in der Tatsache, daß in der industriellen Umgebung die zu erkennenden Teile alle bekannt sind. Es ist grundsätzlich nicht nötig, Objektklassen zu bilden, die die Möglichkeit beinhalten, bisher nicht bekannte Objekte trotzdem einer bekannten Objektklasse zuzuordnen. Allerdings treten das Problem der Fertigungstoleranzen und die speziellen Bildanalyseprobleme auf, die wir weiter unten noch betrachten werden.

Als Vorlage für dieses Beispiel dient ein kleiner Elektromotor, von dem einige Teile in Bild 8.11 zu sehen sind. In der industriellen Umgebung sind die Primelemente, die wir in Kapitel 5 definiert haben, nicht ausreichend. Dies haben auch Lim et al. /LIM 84/ festgestellt, als sie mit dem ACRONYM-System Werkstücke erkennen wollten. Deshalb haben sie Löcher als von Körpern umgebenen

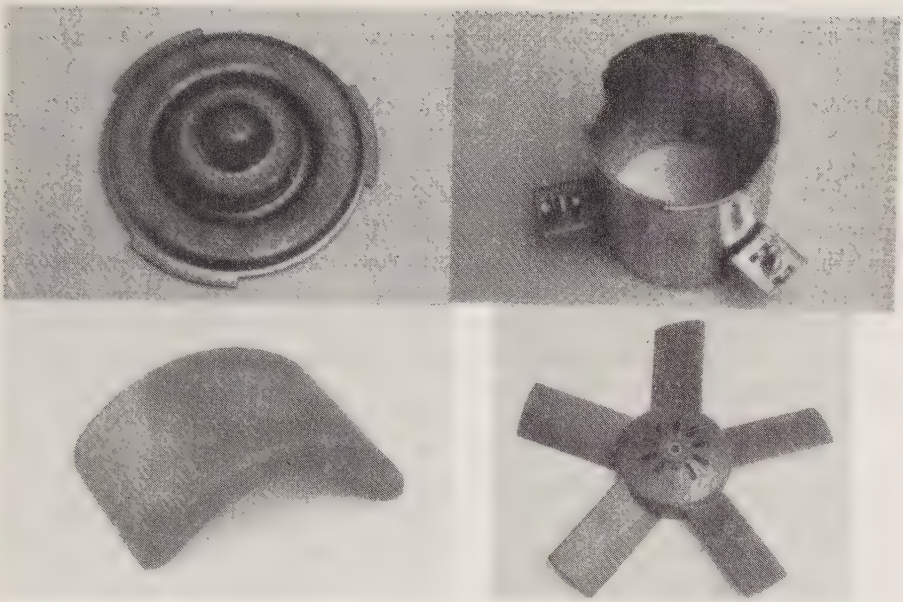


Bild 8.11: Einige Werkstücke

Zylindern, Federn als Zylinder mit einer Helix als Achse und stabile Zustände zusätzlich aufgenommen. Entsprechend der in Kapitel 5 erläuterten Möglichkeit haben wir zwei neue Primelemente definiert /HES 86a/, in dem wir zwei neue Grundflächen RING und TEILRING in das BMS aufgenommen haben. Die zugehörigen VZ (siehe Bild 8.12) ergeben sich dann aus Grundfläche, Deckfläche und Höhe, wobei

Bild 8.12: Verallgemeinerte Zylinder

Grund- und Deckfläche die gleiche Form haben müssen. Ein RING als Grundfläche wird durch die Angabe eines Außenradius (ARADIUS) und eines Innenradius (IRADIUS) definiert. Es gelten die gleichen Einschränkungen wie beim Kreis. Der RING wird von zwei konzentrischen Kreisen erzeugt, deren gemeinsamer Mittelpunkt bei der Grundfläche den Ursprung des lokalen KOS markiert. Wie bei den zylindrischen VZ sind auch hier keine verschobenen Primelemente zugelassen (vgl. Bild 5.4). Die Definition des TEILRING-Primelements erfolgt genauso. Man muß nur noch, getrennt für Grund- und Deckfläche, angeben, wieviele Grad des vollen Rings der TEILRING haben soll. Dabei wird der Kreisbogen (BOGEN) gegen den Uhrzeigersinn abgetragen und beginnt an der x-Achse des lokalen KOS. Eine andere Lage eines TEILRINGS in einem Objekt muß durch die Transformation der KOS ineinander ausgedrückt werden. Die in Abschnitt 5.6 aufgestellte Syntax muß dann folgendermaßen erweitert werden:

```
GRUNDF   ::= (VIERE v KREIS v DREIE v RING v TEILRING v ...) & (BED)*
DECKF    ::= (VIERE v KREIS v DREIE v RING v TEILRING v ...) & (BED)*
RING     ::= ARADIUS & FUAR & IRADIUS & FUIR
TEILRING ::= ARADIUS & FUAR & IRADIUS & FUIR & BOGEN & FUBO
ARADIUS  ::= reelle Zahl
IRADIUS  ::= reelle Zahl
BOGEN    ::= reelle Zahl
```

Die Numerierung der Oberflächen wird gemäß der in Abschnitt 5.3 vereinbarten Konvention vorgenommen. Die Grundfläche trägt bei beiden Primelementen die Nummer 1, und die Deckfläche die höchste Nummer, also 4 bzw. 6. Beim RING hat die Außenfläche die Nummer 2 und die Innenfläche die Nummer 3, während man beim TEILRING mit der lokal in der x-,z-Ebene liegenden Stirnfläche (Nummer 2) zu zählen beginnt: Außenfläche 3, zweite Stirnfläche 4 und Innenfläche 5. Bei der Zusammensetzung zu Objekten kann man mit diesen Primelementen Objekte mit Löchern erzielen, ohne die Operationen Durchschnitt und Differenz zu benötigen.

Die rechnerinterne Behandlung von Toleranzen bei Werkstücken ist bisher noch nicht allgemein gelöst. Einen möglichst umfassenden Ansatz dazu legt Requicha /REQ 83b/ vor. Er kritisiert vor allem, daß bisherige Toleranzberechnungen immer von einem idealen Objekt (z.B. einem idealen Zylinder) ausgehen und für diesen dann die zulässige Abweichung des Radius angeben. Leider liegt aber in der Realität kein idealer Zylinder vor, so daß es schwierig ist, wo und wie die Abweichung gemessen werden muß. Requicha erarbeitet die mathematischen Grundlagen, um Toleranzen in einem CAD-System eindeutig angeben zu können. Darauf aufbauend wurden die Toleranzen in PADL-2 /BRW 82/ definiert. Ohne das gesamte System darlegen zu können, seien Grundzüge am Beispiel eines Kreises erläutert.

Er wird mit drei unabhängigen Toleranzzonen definiert, die jeweils einen Ring erzeugen, innerhalb dessen der reale Kreis liegen muß. Es sind die Größentoleranz, die Formtoleranz und die Lagetoleranz. Erst wenn der Kreis alle drei Toleranztests erfolgreich passiert, gilt er als innerhalb der Toleranz liegend.

Dieser Ansatz hat allerdings zwei Nachteile. Zum einen ist er mehr für die Synthese und für das Zeichnen der Objekte ausgelegt. Er ist ja für Probleme, die beim Computerunterstützten Konstruieren (CAD) auftreten, entwickelt. Gerade deshalb ist er auch sehr umfangreich und detailliert, um alle aus dem technischen Zeichnen bekannten Toleranztechniken zu erfassen. Darum halten wir eine Toleranzbildung mit Fuzzy-Funktionen in einem BMS wegen der Einfachheit trotzdem für geeignet. Außerdem bieten gerade die Fuzzy-Funktionen durch die "vagen" Grenzen eine elegante Möglichkeit, Fehler, die bei der Aufnahme oder während der Bildverarbeitung entstehen, geeignet zu bewerten.

Für zwei der Werkstücke aus Bild 8.11 wurden Objektmodelle generiert. Ein Deckel des Elektromotors (siehe Bild 8.13) ist aus einem Zylinder, sieben Ringen und drei Teilringen zusammengesetzt. Dabei haben die lokalen KOS aller Primelemente eine gemeinsame z-Achse, auf der der Ursprung der verschiedenen

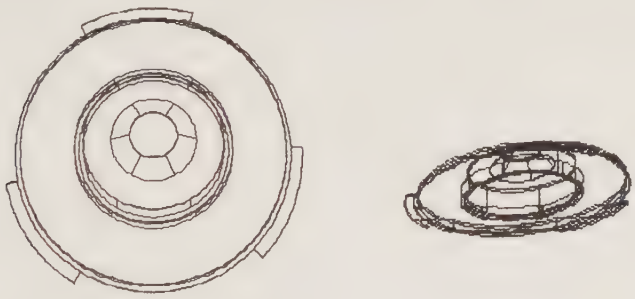


Bild 8.13: Deckel eines Elektromotors

KOS durch eine Translation nur nach oben oder unten verschoben ist. Das Gehäuse des Elektromotors (siehe Bild 8.14) ist aus einem Ring, vier Teilringen und drei Haltern aufgebaut, die jeweils wiederum aus einem Zylinder und zwei Teilringen bestehen. Der Zylinder innerhalb der Halter steht für eine Schraube. Denn Schrauben und Federn sind bisher nur durch Approximation in das Modell einzubringen. Eine ähnliche Vorgehensweise verfolgen Lim et al. /LIM 84/, die Federn approximieren durch den einhüllenden Zylinder und die Angabe von parallelen Linien von einem beliebigen Blickpunkt aus.

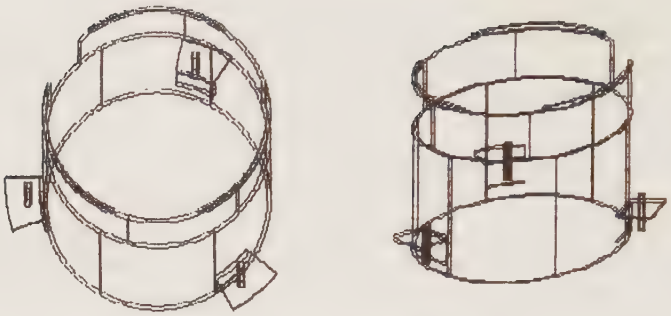


Bild 8.14: Gehäuse eines Elektromotors

Sollen die Werkstücke durch einen Roboter gegriffen werden, so muß ins Modell noch eine Relation Greifpunkte eingefügt werden. Dabei können die Greifpunkte in lokalen 3D-Koordinaten abgelegt werden. Eine weitere Komponente im Modell kann z.B. ein Vorranggraph sein, der für eine Montageaufgabe angibt, in welcher Reihenfolge die Werkstücke zusammengesetzt werden müssen. Das Hinzufügen einer weiteren Relation stellt bei einer relationalen Datenbank kein Problem dar, so daß den Anforderungen, die an das BMS gestellt werden, zumindest in dieser Hinsicht genügt werden kann.

9. Zusammenfassung

Die Probleme, die in der Bildanalyse zu lösen sind, werden immer umfangreicher, so daß in vielen Fällen herkömmliche Systemansätze nicht mehr ausreichen. Entsprechend geht man auch in der Bildanalyse, wie in der gesamten Informatik, von der Informationsverarbeitung zur Wissensverarbeitung über. Mit Schlagwörtern wie "modellgestützte Bildanalyse" oder "wissensbasierte Bildverarbeitung" wird ein Systemansatz beschrieben, in dem ein Modell des Ausschnitts der Welt, mit dem sich das System beschäftigt, explizit gespeichert ist. Man erhofft sich davon zwei wesentliche Vorteile. Zum einen können Verarbeitungsalgorithmen allgemeiner geschrieben werden und dadurch ein umfangreicheres Bild bearbeiten. Die Spezialisierung der Algorithmen geschieht erst durch die Anwendung des im Modell gespeicherten Wissens. Zum anderen kann der Problembereich geändert werden, in dem nur ein kleiner Teil des Systems, nämlich das Modell, ausgetauscht wird, während der Rest des Systems unverändert bleiben kann. Beide Merkmale, eine Vergrößerung des Anwendungsbereichs eines Systems und eine mögliche Veränderung des gesamten Problemkreises, sind für ein allgemeines Bildanalysesystem wichtig.

Bei der Implementierung des Modells eines Bildanalysesystems unterscheiden wir zwei Schritte. Als erstes müssen Modelle der Objekte der realen Welt erstellt werden, die, entsprechend den Anforderungen der Bildanalyse, vor allem geometrische und visuelle Eigenschaften beinhalten. Nur am Rande sind wir darauf eingegangen, daß in umfassenderen Systemen das Modell auch noch andere Informationen, wie z.B. funktionelle und physikalische Merkmale der Objekte, enthalten muß. Doch dies ist sicher ein Ziel für die Zukunft. Heutige Bildanalysesystem gründen ihre Objektmodelle meist noch auf zweidimensionale Merkmale wie Kennzahlen (Fläche, Umfang, umschreibendes Rechteck usw.) oder Kantenstückchen (Geraden, Kreisbögen, Ecken usw.). Dies ist vor allem dort der Fall, wo von der Anwendung her Geschwindigkeit und Genauigkeit als wichtiger angesehen werden als Flexibilität und Allgemeingültigkeit. In Forschungssystemen werden dagegen schon dreidimensionale Modelle erprobt, die sich auf verallgemeinerte Zylinder (VZ) oder dreidimensionale Oberflächen gründen. Bedeutend weiter entwickelt ist die dreidimensionale Modellbildung bei den geometrischen Modellieren der CAD-Systeme. Dort hat sich intern eine Modellbildung mit 3D-Oberflächen oder CSG durchgesetzt, während zur Eingabe die dem Operator anschaulichen Verfahren mit CSG oder mit Symmetriekörpern zum Einsatz kommen. Auch wenn verschiedene Forscher diese CAD-Systeme in die Bildanalyse übertragen wollen, zeigen sich doch die Anforderungen der Bildanalyse von denen der Bildsynthese (CAD und auch

Computeranimation) so unterschiedlich, daß eine eins-zu-eins-Übertragung nicht möglich ist.

Der zweite Schritt bei der Erstellung eines Modells ist die interne Repräsentation. Dieser Bereich wird unter dem Begriff Wissensrepräsentation vor allem in der Künstlichen Intelligenz intensiv untersucht. Dort finden vor allem die vier Formalismen Prädikatenlogik 1. Stufe, Produktionensysteme, Frames und semantische Netze Verwendung. Davon kommen semantische Netze, Produktionensysteme und Frames auch in Bildanalysesystemen zum Einsatz. Diese Wissensrepräsentationsformalismen stellen neben den Datenstrukturen zur Abspeicherung und zum Zugriff auf das Wissen auch Möglichkeiten zu Schlußfolgerungen zur Verfügung. Diese Möglichkeiten gibt es bei relationalen Datenbanken, formalen Grammatiken und Listenstukturen nicht. Trotzdem sind diese Formalismen in vielen Bildanalysesystemen vertreten, da sie einfach sind und für die Zwecke der Bildinterpretation ausreichen. Im ganzen ist die geeignete Wahl des Wissensrepräsentatitonsformalismus vor allem für die Effizienz des Systems zuständig. Für den Umfang und die Genauigkeit der Bildanalyse zeichnet vor allem die Modellbildung verantwortlich.

Nach diesem Überblick wird ein Bildanalysemodellierungssystem vorgestellt, das Strukturen für die Modellbildung und die interne Darstellung der Objektmodelle beinhaltet.

Grundbausteine des Modellierungssystems sind verallgemeinerte Zylinder (VZ). Als Grund- und Deckfläche dieser VZ sind dabei Vierecke, Kreise, Ringe und Teile von Ringen zugelassen. Eine Erweiterung dieser Elemente ist möglich. Ein VZ wird durch zwei parallele Flächen (Bedingung: gleiche Form) als Grund- und Deckfläche sowie die Angabe seiner Höhe beschrieben. Jedem VZ wird ein lokales Koordinatensystem zugeordnet. Objekte ergeben sich durch die Vereinigung solcher Grundbausteine. Diese Vereinigung wird erklärt als Transformation der lokalen Koordinatensysteme in ein Objektkoordinatensystem, das einem VZ dem sog. Hauptteil, zugeordnet ist. Weiterhin ist die Bildung von Objektklassen möglich. Hierzu werden verschiedene Parameter der Objektmodelle mittels Fuzzyunktionen variabel definiert. Für die Bildanalyse notwendige Oberflächenattribute, w. z. B. Farbe, Intensität oder Textur, können zusätzlich abgelegt werden, um für die Interpretation eines Bildes zur Verfügung zu stehen. Aus mehreren Objektmodellen kann eine Objektgruppe gebildet werden, wenn ein gemeinsames Auftreten dieser Objekte sehr wahrscheinlich ist. Außerdem ist die Definition von Regionen für solche Objekte möglich, die nicht oder nur schwierig dreidi-

mensional modelliert werden können (z. B. Himmel in einer Vorstadtszene). Das gesamte Modell setzt sich dann aus Regionen, Objektklassen, Objektgruppen und deren mögliche örtliche Beziehungen zusammen. Zur Abspeicherung des Modells dient eine relationale Datenbank, die Relationen der ersten Normalform enthält. Für ein eingeschränktes Modell einer natürlichen Szene sind hier 15 Relationen aufgebaut, die sowohl die grundlegenden Daten der Objektmodelle als auch Hilfsdaten enthalten. Der Vergleich Bilddaten - Modell erfolgt auf der Ebene der VZ durch die Auswertung der Fuzzyfunktionen, die die Parameter des entsprechenden VZ definieren.

Dieser Ansatz wird nochmals an einer Methode überprüft, die aus den im CAD bekannten Verfahren zur Modellbildung entwickelt wurde. Hierbei wird ein CSG-Baum mit Primelementen verwendet, die in Oberflächenzerlegung modelliert sind. Dazu kommen die für die Bildanalyse wichtigen Eigenschaften der Oberflächenbeschreibung und der Objektklassenbildung. Ein Vergleich der beiden Methoden favorisiert den Ansatz mit den VZ, da dieser für sinnvoller gehalten wird und außerdem besser ausgearbeitet ist.

Da zum Test des BMS kein Bildanalysesystem zur Verfügung stand, wurden zwei Beispielproblemkreise betrachtet und ein Kontrollalgorithmus konzipiert, um zumindest eine Interpretation einmal durchzuführen. Der Kontrollalgorithmus arbeitet objektmodellbezogen, d.h. er wählt datengetrieben ein Objektmodell aus, das dann interpretiert wird. Dazu werden alle Teile des Objektmodells hypothetisiert und mittels A^*-Algorithmus eine Interpretation der Teile durch Strukturen aus dem Bild versucht. Ist ein Objektmodell interpretiert, wird eine weitere noch nicht bekannte Struktur aus dem Bild gewählt und das nächste Objektmodell hypothetisiert. Zum Ende des Algorithmus müssen noch die im Bild vorhandenen Regionen interpretiert werden.

Die derzeitige Modellerstellung wird von dem Operator oder Experten geprägt, der sie durchführt. Der gesamte Prozeß der Ableitung der Modelle aus den Objekten des Problemkreises wird noch vom Bediener erledigt. Er findet nur bei der Visualisierung und beim Eintrag in die relationale Datenbank Unterstützung. Dies ist einer der Punkte, die in Zukunft erforscht werden müssen, da die Erstellung eines einigermaßen umfänglichen Modells sehr aufwendig ist. Das erste Beispiel zeigt einen Blick aus dem Fenster. Es umfaßt drei Testbilder, von denen zwei synthetisch erzeugt wurden, d.h. der Operator hat VZ und Regionen aufgrund von möglichen Objektmodellen zusammengestellt und so ein Testbild erzeugt. Das dritte Testbild wurde aus einem Foto entwickelt, in dem der Opera-

tor die Bildverarbeitung simuliert und so VZ und Regionen extrahiert hat. Die Interpretation mittels des Kontrollalgorithmus zeigt, daß das BMS unter den gegebenen Bedingungen geeignet ist, in einem Bildanalysesystem zu arbeiten. Die Interpretation ist fast vollständig richtig und aufgrund der aus dem Modell extrahierten Informationen schneller und genauer. Es zeigt sich sogar, daß noch weitere Fehler vermieden werden könnten, wenn weiteres im Modell abgelegtes Wissen in die Interpretation miteinbezogen würde.

Bei den industriellen Szenen (Werkstücken) fällt auf, daß die sehr stark gegenüber den natürlichen Szenen geänderten Anforderungen die Definition von zusätzlichen Primelementen notwendig machen. Dies ist zwar möglich und erfolgt, aber hier zeigt sich ein Problem der Modellbildung mit VZ. Diese sind immer dann besonders gut, wenn der Problemkreis eher allgemein gehalten ist und variable, unbekannte Objekte enthält. Bei der Modellierung für wenige, bekannte, aber detaillierte Objekte stoßen die VZ an ihre Grenzen (begrenzter Definitionsbereich). So sind Meßaufgaben damit schwierig zu lösen. Allerdings für typische Bildanalyseprobleme mit Aufnahmefehlern u.ä. könnte sich das vorgestellte BMS mit einigen Erweiterungen und den Objektklassen auf der Basis von Fuzzy-Funktionen durchaus eignen.

Dreidimensionale Modellbildung wird sich überall dort durchsetzen, wo keine spezifische Vorhersage über die Ansicht eines Objekts gemacht werden kann. Denn nur dadurch ist die Standpunktunabhängigkeit gewährleistet. Auch die explizite Speicherung des Modells wird in Zukunft wegen der gewünschten Flexibilität der Systeme häufiger angestrebt werden. VZ bilden einen Ansatz zur Bildung von dreidimensionalen Objektmodellen. Ihr Nachteil liegt vor allem beim Defintionsbereich und bei der Zugänglichkeit. Der Definitionsbereich ist bei natürlichen und bei vom Menschen erzeugten Objekten immer dann ausreichend, wenn der Detaillierungsgrad nicht zu groß ist und wenn keine außergewöhnlichen Oberflächen auftreten. Die Abbildung von 3D- auf 2D-Daten muß bei VZ explizit in irgendeiner Form berechnet werden, sei es in einem Vorlauf wie bei ACRONYM, sei es direkt bei der Interpretation. Dies kann ziemlich aufwendig werden. Demgegenüber bieten VZ den Vorteil, daß sie einfach erzeugt werden können, da sie eine dem Menschen durchaus vertraute Darstellungsweise sind. Sie sind gut zu handhaben, da sie prägnant sind und sie unterstützen Objektklassenbildung und Hierarchisierung, was für die Bildanalyse von entscheidendem Vorteil ist. Deshalb finden sie in einigen Bildanalysesystemen Verwendung und werden auch in dieser Arbeit eingesetzt.

Die Grundlagen der Modellbildung mit VZ wurden in dieser Arbeit aufgezeigt. Ein wichtiger Schritt ist jetzt die Einbindung des BMS in ein Bildanalysesystem. Dazu müßte neben den übrigen Komponenten des Systems der Kontrollalgorithmus eine Verbesserung erfahren und die genaue Schnittstelle zwischen Bildverarbeitung und Modell nochmals überlegt werden. Erst in einem solchen System ist eine endgültige Beurteilung dieses Ansatzes möglich. Eine andere interessante Perspektive ist die automatisierte Wissensakquisition. Die Frage, wie weit man den Prozeß der Modellerstellung automatisieren kann, ist bisher noch ungeklärt und wird erst in Anfängen untersucht.

Literatur

/ADO 84/ Adorni, G., Di Manzo, M., Giunchiglia, F.
From Description to Images: What Reasoning in Between ?
Proc. ECAI 1984, Pisa, pp. 359 - 368

/AGI 73/ Agin, G., Binford, Th.
Computer Description of Curved Objects
Proc. 3rd IJCAI, 1973, pp. 629-640

/AGI 76/ Agin, G., Binford, Th.
Computer Descriptions of Curved Objects
IEEE Trans Comp. vol 25, 1976, pp. 439-449

/AGI 80/ Agin, G.
Computer Vision Systems for Industrial Inspection and Assembly
IEEE Computer, may 1980, pp. 11-20

/AGI 81/ Agin, G.
Hierarchical Representation of Three-Dimensional Objects
Using Verbal Models
IEEE Trans. PAMI-3, mar. 1981, pp. 197 - 204

/BAJ 74/ Bajcsy, R. K., Tavakoli, M.
Computer Recognition of Roads from Satellite Pictures
2nd IJCPR, 1974, pp. 190 - 194

/BAL 78/ Ballard, D., Brown, C., Feldman, J.
An Approach to Knowledge-Directed Image Analysis
Aus : A.Hanson,E.Riseman Computer Vision Systems
Acadamic Press 1978

/BAL 82/ Ballard, D., Brown, C.
Computer Vision
Prentice Hall, 1982

/BAR 81/ Barr, A., Feigenbaum, E.
The Handbook of Artificial Intelligence Vol. 1
Pitman Books, 1981

/BAS 86/ Barsky, B.
Parametric Spline Curves and Surfaces
IEEE Comp. Graph. Appl. vol 6, no. 2, feb. 1986, pp. 33-64

/BEE 80/ Bense, H.
Modularer Datenbank Entwurf fuer Kleinrechner
GI - Jahrestagung, 1980, Springer-Verlag, S. 225 - 242

/BEN 85/ Benn, W.
Toleranter Vergleich von Strukturen mit erweiterten, nicht-normalisierten Relation
Proc. Mustererkennung 1985, Erlangen 1985, Springer Verlag

/BES 85/ Besl, P., Jain, R.
Three-Dimensional Object Recognition
Computing Survey, vol. 17, no. 1, mar. 1985, pp. 75-145

/BEZ 81/ Bezdek, J. C.
Pattern Recognition with Fuzzy Objective Function Algorithms
Plenum Press, 1981

/BHA 85/ Bhanu, B., Henderson, T.
CAGD Based 3-D Vision
IEEE Int. Conf. Rob. and Automation, St. Louis, 1985, pp. 411 - 417

/BIN 82/ Binford, Th.
Survey of Model-Based Image Analysis Systems
Int. Journal of Robotics Research Vol. 1, no.1, 1982

/BOL 83/ Bolles, R., Horand, P., Hannah, M.
3DPO: A Three-Dimensional Part Orientation System
Proc. IJCAI 1983, Karlsruhe, pp. 1116-1120

/BRD 83/ Brady, M.
Criteria for Representation of Shape
In: Beck, J., et al.: Human and Machine Vision, Academic Press, 1983

/BRO 81/ Brooks, R.
Symbolic Reasoning Among 3-D Models and 2-D Images
Artificial Intelligence, vol. 17 (1981), pp. 285-348

/BRW 81/ Brown, C.
Some Mathematical and Representational Aspects of Solid Modeling
IEEE PAMI, vol. 3, no. 4, jul. 1981, pp. 444-453

/BRW 82/ Brown, C.
PADL-2: A Technical Summary
IEEE Comp. Graph. Appl. vol. 2, no. 2, 1982, pp. 69-84

/BUC 84/ Buchanan, B., Shortliffe, E.
Rule-Based Expert Systems
Addison-Wesley, 1984

/BUN 83/ Bunke, H., Sagerer, G., Niemann, H.
Model Based Analysis of Scintigraphic Image Sequences of the Human Heart
Aus : Huang, T. S., Image Sequence Processing, Springer Verlag, 1983

/BUN 85/ Bunke, H.
Modellgesteuerte Bildanalyse
Teubner Verlag, 1985

/CAL 83/ Calla, G. Mc, Cercone, N.
Guest Editors' Introduction: Approaches to Knowledge Representation
Computer, vol. 16, no. 10, oct. 1983, pp. 12-18

/CAT 84/ Castore, G., Crawford, C.
From Solid Model to Robot Vision
Int. Conf. Robotics, Atlanta, 1984, pp. 90-92

/CHE 81/ Cheng, J., Huang, T.
Image Recognition by Matching Relational Structures
Proc. PRIP 81, Dallas, pp.542-547

/CHL 86/ Chalmers, A. F.
Wege der Wissenschaft
Springer Verlag, 1986

/COO 85/ Cooper, B.
CAD/CAM-Einsatz bei der Konstruktion und Fertigung von hydraulischen Schaltelementen
CAD/CAM 6/86, S. 62-66

/DAT 81/ Date, C. J.
An Introduction to Database Systems
Addison-Wesley, 1981

/DEK 83/ Deken, J.
Computer Images State of the Art
Stewart, Tabori & Chang Pub, N. Y., 1983

/EBL 84/ Eberlein, W.
Architektur Technischer Datenbanken für Integrierte Ingenieursysteme
Arbeitsberichte des IMMD, Bd. 17, no. 1, Erlangen 1984

/EIC 86/ Eichhorn, W., Niemann, H.
A Birectional Control Strategy in a Hierarchical Knowledge Structure
Proc. 8th ICPR, Paris, 1986

/FAI 84/ Fairchild
CCD - The Solid State Imaging Technology
Fairchild 1984

/FIC 85/ Fichte, H.
Implementierung einer relationalen Datenbank auf dem Mikrocomputer-system intel SYS 86/330 (Benutzerschnittstelle und Datensystem)
Diplomarbeit, Lehrstuhl für Informatik 5 (Mustererkennung), Universität Erlangen-Nürnberg, Erlangen 1985

/FOL 82/ Foley, J. D., Dam, A. van
Fundamentals of Interactive Computer Graphics
Addison - Wesley, 1982

/FRE 74/ Freeman, H.
Computer Processing of Line-Drawing Images
acm Computing Surveys vol. 6, 1974, pp. 57-97

/FU 82/ Fu, K. S.
Syntactic Pattern Recognition and Applications
Prentice Hall 1982

/GOB 85/ Goebel, M.
Definition von Ansaetzen zur geometrischen Modellierung fuer die Bildanalyse
Diplomarbeit, Lehrstuhl fuer Informatik 5 (Mustererkennung), Universitaet Erlangen - Nuernberg, 1985

/GON 77/ Gonzales, R., Wintz, P.
Digital Image Processing
Addison Wesley, 1977

/GRE 84/ Grebner, K.
Strukturelle Analyse von Volumenkurven
Diplomarbeit, Lehrstuhl für Informatik 5 (Mustererkennung),
Universität Erlangen-Nürnberg, Erlangen 1984

/GRE 85/ Grebner, K.
Modellgesteuerte Bildanalyse am Beispiel industrieller Szenen
Mustererkennung 1985, Proc., Erlangen, Springer, S. 42 - 46

/HAA 85/ Haralick, R., Shapiro, L.
Image Segmentation Techniques
Comp. Vision, Graph.,Im. Proc. vol. 29, no. 1, 1985, pp. 100-132

/HAE 78/ Haerder, T.
Implementierung von Datenbank-Systemen
Hanser Verlag 1978

/HAN 78/ Hanson, A., Riseman, E.
VISIONS: A Computer System for Interpreting Scenes
From: Hanson, A., Riseman, E. Computer Vision Systems
Academic Press 1978

/HEN 85/ Henderson, T., Bhanu, B.
Intrinsic characteristics as the Interface between CAD and Machine
Vision Systems
Pattern Recognition Letters 3, 1985, pp. 425-430

/HES 84/ Hess, P.
Modellbildung und Wissensdarstellung in Bildanalysesystemen
Interner Bericht, Lehrstuhl fuer Informatik 5 (Mustererkennung),
Universitaet Erlangen - Nuernberg, 1984

/HES 85/ Hess, P., Goebel, M.
Zur Modellierung von natuerlichen Szenen fuer die Bildanalyse
Mustererkennung 1985, Proc., Erlangen, Springer, S. 251 - 255

/HES 86a/ Hess, P.
Three-dimensional modeling of industrial parts for image analysis
Appl. Art. Int. IV, Gilmore, J. (ed.), Proc. SPIE 657, pp. 164-170

/HES 86b/ Hess, P., Kordes, Chr.
Modellgestützte Kontrolle in der Bildanalyse
Mustererkennung 1986, Proc., Paderborn, 1986, Springer Verlag

/HOF 82/ Hofmann, I., Otto, M.
Ein Kontrollalgorithmus für die Verwendung von semantischen
Netzwerken zur Bildanalyse
Diplomarbeit, Lehrstuhl für Informatik 5 (Mustererkennung),
Universität Erlangen-Nürnberg, Erlangen 1982

/HOF 86/ Hofmann, I., Niemann, H., Sagerer, G.
Model Based Interpretation of Image Sequences from the Heart
Proc. Pattern Recognition in Practice II 1985, Amsterdam, North
Holland 1986

/HUL 82/ Huelsmeier, H., Huelsmeier, M.
Entwurf und Implementation eines relationalen Datenbanksystems auf einem Mikrorechner
Diplomarbeiten, Lehrstuhl fuer Informatik 6,
Universitaet Erlangen - Nuernberg, 1982

/HUM 83/ Hummel, R., Zucker, S
On the foundation of relaxation labeling processes
IEEE PAMI vol. 5, may 1983, pp. 267-287

/INH 83/ Inhetveen, R., Luft, A. L.
Abstraktion, Idealisierung und Modellierung bei der Spezifikation, Konstruktion und Verifizierung von Software-Systemen
Angewandte Informatik 12/83, S. 541-548

/INH 86/ Inhetveen, R.
Ueber den Sinn des Gebrauchs von Modellen
Aus : Gatzemeier (Hrsg.) Aachener Beitraege zur Philosophie 1986

/KOR 86/ Kordes, Chr.
Konzeption und Realisierung eines Kontrollalgorithmus für ein Bildanalysesystem
Diplomarbeit, Lehrstuhl fuer Informatik 5 (Mustererkennung),
Universitaet Erlangen - Nuernberg, 1986

/KOE 86/ Koetter, R.
Modell und Oekonomische Realitaet
Aus : Jahrbuch Oekonomie und Gesellschaft
"Die Neoklassik und ihre Kritik", 1986

/LAN 86/ Langenheder, W.
Warum Computer weder hoeren noch sehen auch nicht sprechen, geschweige denn entscheiden koennen
Computer magazin 1,2/86, S. 14-18

/LAU 86/ Laurer, J.
Konzeption und Realisierung eines interaktiven Modelleingabewerkzeugs
Studienarbeit, Lehrstuhl fuer Informatik 5 (Mustererkennung),
Universitaet Erlangen - Nuernberg, 1986

/LEV 78/ Levine, M. D.
A Knowledge-Based Computer Vision System
From: Hanson, A., Riseman, E. Computer Vision Systems
Academic Press 1978

/LEV 81/ Levine, M., Shaheen, S.
A Modular Computer Vision System for Picture Segmentation and Interpretation
IEEE Tr. Patt. Anal. Mach. Intel. 3, sep. 1981, pp. 540-556

/LEV 85/ Levine, M.
Vision in Man and Machine
Mc Graw-Hill, 1985

/LIM 84/ Lim, H., Cowan, C., Chelberg, D.
ACRONYM Model Based Vision in the Intelligent Task Automation Project
SPIE vol. 507, Proc. and Disp. of 3D Data II, 1984, pp. 66 - 72

/LOB 80/ Lobregt, S., Verbeek, P., Groen, F.
3D-Skeletonization: Principle and Algorithm
IEEE PAMI vol. 2, 1980, pp. 75-77

/LOR 75/ Lorenzen, P., Schwemmer, O.
Konstruktive Logik, Ethik und Wissenschaftstheorie
B. I. Wissenschaftsverlag, B. I. Taschenbuch 700, 1975

/LOW 80/ Lowerre, B., Reddy, R.
The HARPY Speech Understandig System
From: Lea, W. (ed.), Trends in Speech Recognition, Prenctice Hall, 1980, pp. 340-360

/LUF 84/ Luft, A. L.
Zur Bedeutung von Modellen und Modellierungsschritten in der Softwaretechnik
Angewandte Informatik 5/84, S. 189-196

/MAA 83/ Matra Datavision
EUCLID
Firmenprospekt, 1983

/MAR 82a/ Marr, D.
Visual Information Processing: The Structure and Creation of Visual Representation
From: Albrecht, D. Recognition of Pattern and Form
Proc. Austin 1979, Springer Verlag, 1982

/MAR 82b/ Marr, D.
Vision
W. H. Freeman and Comp., 1982

/MIN 75/ Minsky, M.
A Framework for Representing Knowledge
From: Winston, P. The Psychology of Computer Vision
McGraw-Hill, 1975, pp. 211 - 277

/NAG 80/ Nagao, M., Matsuyama, T.
A Structural Analysis of Complex Aerial Photographs
Plenum Press, 1980

/NAG 82/ Nagao, M.
Control Strategies in Pattern Analysis
Proc. 6th ICPR, Muenchen, 1982, pp. 996-1006

/NAL 85a/ Nagel, H.-H.
Analyse und Interpretation von Bildfolgen
Informatik-Spektrum Band 8, 1985, S. 178-200 und S. 312-327

/NEE 85/ Nees, G.
Expertensysteme für die Mustererkennung - Stand und Aussichten
Proc. Mustererkennung 1985, Springer Verlag, 1985, S. 138-158

/NEM 79/ Newman, W., Sproull, R.
Principles of Interactive Computer Graphics
Mc Graw-Hill, Kogakusha 1979

/NIE 80/ Niemann, H.
Mustererkennung - Anwendungen
Informatik-Spektrum 3, 1980, S. 19-30

/NIE 81/ Niemann, H.
Pattern Analysis
Springer Verlag, 1981

/NIE 83a/ Niemann, H.
Klassifikation von Mustern
Springer Verlag, 1983

/NIE 83b/ Niemann, H.
Control Strategies in Image and Speech Understanding
GWAI-83, Dassel, 1983, pp. 31 - 50

/NIE 85a/ Niemann, H.
Wissensbasierte Bildanalyse
Informatik-Spektrum 8, H. 4, Aug. 1985, S. 201-214

/NIL 82/ Nilsson, N.
Principles of Artificial Intelligence
Springer Verlag, 1982

/NIS 81/ Nishihara, H. K.
Intensity, Visible Surface, and Volumetric Representations
Artificial Intelligence, vol. 17, 1981, pp. 265-285

/OHT 84/ Ohta, Y.
Knowledge-based Interpretation of Outdoor Natural Color Scenes
Pitman Books 1985

/OTT 86/ Ott, R.
Sehen, hören, sprechen
elektronikpraxis Nr. 4, April 1986, S. 48-58

/PAV 82/ Pavlidis, T.
Algorithms for Graphics and Image Processing
Springer Verlag, 1982

/PER 78/ Perkins, W. A.
A Model-Based Vision System for Industrial Parts
IEEE Trans. on Computer, vol.C-27, 1978, pp. 126 - 143

/PER 84/ Perkins, W. A.
Model-Based Inspection System for Component Boards
Pattern Recognition, vol. 17, no. 1, 1984, pp. 135 - 140

/PRI 85/ Price, K.
Relaxation Matching Techniques - A Comparison
IEEE PAMI 7, no. 5, 1985, pp. 617-623

/PUP 86/ Puppe, F.
Expertensystem
Informatik-Spektrum Bd. 9, 1986, S. 1-13

/RAD 84/ Radig, B.
Image Sequence Analysis Using Relational Structures
Pattern Recognition vol. 17, no. 1, 1984, pp. 161-167

/RAD 85/ Radig, R., Schlieder, C.
Modellierung symmetrischer Werkstücke
Robotersysteme 1, S. 35-42, 1985

/REI 85/ Reisin, F.-M., Wilhelm, R.
Präzisionsgelenkte Munition
Aus: Bickenbach, J. et. al. (Hrsg.), Militarisierte Informatik,
Berlin 1985, S. 113-120

/REQ 80/ Requicha, A. A. G.
Representation of Rigid Solids: Theory, Methods, and Systems
Computing Surveys 12, Dec. 1980, pp 437-464

/REQ 82/ Requicha, A. A. G., Voelcker, H. B.
Solid Modeling: A Historical Summary and Contemporary Assessment
IEEE Comp. Graph. Appl. vol. 2, no. 2, mar. 1982, pp. 9 - 24

/REQ 83a/ Requicha, A. A. G.
Toward a Theory of Geometric Tolerancing
Int. J. Rob. Research, vol. 2, no. 4, winter 1983, pp. 45 - 60

/REQ 83b/ Requicha, A. A. G., Voelcker, H. B.
Solid Modeling: Current Status and Research Directions
IEEE Comp. Graph. Appl., oct. 1983, pp. 25-37

/ROS 76/ Rosenfeld, A., Hummel, R., Zucker, S.
Scene labeling by relaxation operations
IEEE Syst. Man Cybern. vol. 6, june 1976, pp. 420-453

/ROU 79/ O´Rourke, J., Badler, N.
Decomposition of Three-Dimensional Objects into Spheres
IEEE PAMI vol. 1, no. 3, 1979, pp. 295-305

/RTH 81/ Rosenthal, D.
An Inquiry Driven Vision System Based on Visual and
Conceptual Hierarchies
UMI Research Press 1981

/RUB 78/ Rubin, St.
The ARGOS Image Understanding System
Ph.D. Thesis Carnegie Mellon University 1978

/RUM 84/ Rummel, P., Beutel, W.
Workpiece Recognition and Inspection by a Model-Based
Scene Analysis System
Pattern Recognition, vol. 17, no.1, 1984, pp. 141 - 148

/SAG 85/ Sagerer, G.
Darstellung und Nutzung von Expertenwissen fuer ein Bildanalysesystem
Springer Verlag, 1985

/SAM 84/ Samet, H.
The Quadtree and Related Hierarchical Data Structures
acm Computing Surveys vol. 16, 1984, pp. 187-260

/SCF 85/ Schraft, R. D., Melchior, K. W., Rueff, M.
Musterverarbeitende Systeme in der Prüf- und Robotertechnik
Mustererkennung 1985, Proc., Erlangen 1985, Springer

/SCN 75/ Schneider, H.
Compiler
Walter de Gruyter, Berlin, 1975

/SCO 85/ Schott, K.
Implementierung einer relationalen Datenbank auf dem Mikrocomputer-
system intel SYS 86/330 (Zugriffssystem und Speichersystem)
Diplomarbeit, Lehrstuhl für Informatik 5 (Mustererkennung),
Universität Erlangen-Nürnberg, Erlangen 1985

/SEI 77/ Seitzer, D.
Elektronische Analog-Digital-Umsetzer
Springer Verlag 1977

/SHA 80/ Shani, U.
A 3-D Model-Driven System for the Recognition of Abdominal
Anatomy from CT Scans
Proc. 5th IJCPR, 1980, pp 585-591

/SHF 85/ Schärf, R.
Modellgesteuerte Lokalisierung von Objekten
Proc. 7. DAGM-Symposium Erlangen 1985, Springer Verlag, S. 256-261

/SHI 78/ Shirai, Y.
Recognition of Real-World Objects Using Edge Cue
From: Hanson, A., Riseman, E. Computer Vision Systems
Academic Press 1978

/SHP 84/ Shapiro, L. G.
Relational Matching - Problems, Techniques, and Applications
Mustererkennung 1984, Proc., Graz, Springer Verlag, p. 24-41

/SHP 85/ Shapiro, L., Haralick, R.
A Metric for Comparing Relational Descriptions
IEEE PAMI, vol. 7, no. 1, jan. 1985, pp. 90 -94

/SIE 86/ Sielaff, C.
Hierarchische Dekomposition und Synthese von Objekten
GWAI 85, Springer Verlag, 1986, S. 348-355

/SIM 85/ Sim, R.
Recent Developments in Computer Aided Design and Engineering
CAD International Directory 1986, Butterworth, 1985

/SPU 84/ Spur, G., Krause, F.-L.
CAD-Technik
Carl Hauser Verlag, München, 1984

/SRI 81/ Srihari, S.
Representation of Three-Dimensional Digital Images
Computing Surveys, vol. 13, no. 4, dec. 1981, pp. 399 - 424

/STI 86/ Stein, G., Topf, H., Walter, I.
Interpretation of Insdustrial Scenes Using Augmented Transition Networks and Production Systems: Procedural vs. Declarative Models
Proc. Pattern Recognition in Practice II 1985, North Holland 1986

/TAN 80/ Tanimato, S., Klinger, A. (eds.)
Stuctured Computer Vision
New Yor, Academic Press, 1980

/TEN 74/ Tennenbaum, J., Garvey, T., Weyl, Wolf
ISIS: An Interactive Scene Interpretation System
Proc. 2nd IJCPR, 1974, pp. 123 - 125

/THA 83/ Magnenat-Thalmann, N., Thalmann D.
The Use of High-Level 3-D Graphical Types in the Mira Animation System
IEEE Comp. Graph. & Appl., vol. 3, dec. 1983, pp. 9-15

/THA 85/ Magnenat-Thalmann, N. et. al.
MIRA-Shading: A Structured Language for the Synthesis and Animation of Realistic Images
From: Kunii, T. (ed.), Frontiers in Computer Graphics, Tokyo, Springer Verlag 1985

/TSI 83/ Tsipis, K.
Marschflugkörper
Aus: Rüstung und Abrüstung, Spektrum der Wissenschaft, Heidelberg, 1983

/WAN 83/ Wang, H., Sun, H., Yada, S., Rosenfeld, A.
Some experiments in relaxation image matching using corner features
Pattern Recog., vol. 16, 1983, pp. 167-182

/WED 76/ Wedekind, H., Härder, T.
Datenbanksysteme II
BI-Wissenschaftsverlag, Mannheim, 1976

/WED 78/ Wedekind, H.
Relationale Datenbanksysteme
Informatik-Spektrum 1, 1978, S. 5 - 16

/WIK 85/ Winkler, G.
Industrielle Anwendung der digitalen Bildverarbeitung
Informatik-Spektrum 8, H. 4, Aug. 1985, S. 215-224

/YAC 77/ Yachida, M., Tsuji, S.
A Versatile Machine Vision System for Complex Industrial Parts
IEEE Tr. Computers, vol. 26, 1977, pp. 882 - 894

/ZAD 75/ Zadeh, L. et al.
Fuzzy Sets and their Application to Cognitive and Decision Processes
Academic Press, 1975

/ZAD 79/ Zadeh, L.
A Theory of Approximate Reasoning
Hayes, J., Mitchie, D., Mijtlich, L. (eds.), Machine Intelligence 9, 1979, pp. 149-194

/ZAD 85/ Zadeh, L.
A Formalization of Commonsense Reasoning Based on FUZZY Logic
From: Wissensbasierte Systeme, GI-Kongreß, 1985,
Springer Verlag, pp. 398-402

ANHANG

Anhang A: Relationen des Modells 136

Anhang B: Relationen des Bildes 142

Anhang C1: Lokale Bewertungen mit erhöhtem Minimum 145

Anhang C2: Lokale Bewertungen mit Mittelwert 148

**** RELATIONENKONTROLLBLOCK :BILD_K .RCB ****

ATTRIBUTNAME	FORMAT	SCHLUESSEL
NAME	3	P
HAUPTOBJ	3	N
OBJEKT	3	P
MODAL	1	N
ZUSHANG	3	N

**** RELATIONENKONTROLLBLOCK :EINZELOBJK.RCB ****

ATTRIBUTNAME	FORMAT	SCHLUESSEL
NAME	3	P
HAUPTTEIL	3	N
TEILE	3	P
MAXANZ	1	N
NOTWENDIG	1	N

**** RELATIONENKONTROLLBLOCK :FUZZY .RCB ****

ATTRIBUTNAME	FORMAT	SCHLUESSEL
NAME	3	P
PUNKT_1	2	N
PUNKT_2	2	N
PUNKT_3	2	N
PUNKT_4	2	N

```
**********************************************************************

 ****  RELATIONENKONTROLLBLOCK :GRUNDFLAEB.RCB ****

**********************************************************************

ATTRIBUTNAME      FORMAT      SCHLUESSEL
 ---------------------------------------

NAME                3             P
BED_1               4             N
PAR_11              3             N
PAR_12              3             N
PAR_13              3             N
BED_2               4             N
PAR_21              3             N
PAR_22              3             N
PAR_23              3             N
BED_3               4             N
PAR_31              3             N
PAR_32              3             N
PAR_33              3             N
BED_4               4             N
PAR_41              3             N
PAR_42              3             N
PAR_43              3             N
**********************************************************************

**********************************************************************

 ****  RELATIONENKONTROLLBLOCK :GRUNDFLAEK.RCB ****

**********************************************************************

ATTRIBUTNAME      FORMAT      SCHLUESSEL
 ---------------------------------------

NAME                3             P
FORM                3             SO
LOK_DURCH           2             N
FUZZY_DU            3             N
LOK_X1              2             N
FUZZY_X1            3             N
LOK_Y1              2             N
FUZZY_Y1            3             N
LOK_X2              2             N
FUZZY_X2            3             N
LOK_Y2              2             N
FUZZY_Y2            3             N
LOK_X3              2             N
FUZZY_X3            3             N
LOK_Y3              2             N
FUZZY_Y3            3             N
LOK_X4              2             N
FUZZY_X4            3             N
LOK_Y4              2             N
FUZZY_Y4            3             N
**********************************************************************
```

```
**********************************************************************

 ****  RELATIONENKONTROLLBLOCK :OBERFARBEK.RCB ****

**********************************************************************

ATTRIBUTNAME      FORMAT      SCHLUESSEL
 ---------------------------------------

NAME                3             P
ROT                 1             N
FUZZY_RO            3             N
GRUEN               1             N
FUZZY_GR            3             N
BLAU                1             N
FUZZY_BL            3             N
**********************************************************************

**********************************************************************

 ****  RELATIONENKONTROLLBLOCK :OBJBED_K  .RCB ****

**********************************************************************

ATTRIBUTNAME      FORMAT      SCHLUESSEL
 ---------------------------------------

NAME                3             P
EINTRAG             1             P
BED_1               4             N
PAR_11              3             N
PAR_12              3             N
PAR_13              3             N
PAR_14              3             N
BED_2               4             N
PAR_21              3             N
PAR_22              3             N
PAR_23              3             N
PAR_24              3             N
BED_3               4             N
PAR_31              3             N
PAR_32              3             N
PAR_33              3             N
PAR_34              3             N
**********************************************************************

**********************************************************************

 ****  RELATIONENKONTROLLBLOCK :OBJGRUPPEK.RCB ****

**********************************************************************

ATTRIBUTNAME      FORMAT      SCHLUESSEL
 ---------------------------------------

NAME                3             P
HAUPTOBJ            3             N
OBJEKT              3             P
MODAL               1             N
ZUSHANG             3             P
**********************************************************************
```

```
**********************************************************************

 ****  RELATIONENKONTROLLBLOCK :OBJTEILE_K.RCB ****

**********************************************************************

ATTRIBUTNAME      FORMAT      SCHLUESSEL
 ---------------------------------------

NAME                3             P
PRIMKOER            3             SO
OBERFARBE           3             S1
INTENSIT            1             S2
FUZZY_INT           3             N
OBERTEXT            3             N
OBERNUMM            1             P
**********************************************************************
**********************************************************************

 ****  RELATIONENKONTROLLBLOCK :SWEEPINGK .RCB ****

**********************************************************************

ATTRIBUTNAME      FORMAT      SCHLUESSEL
 ---------------------------------------

NAME                3             P
CHARAKT             3             N
STEIGUNG            2             N
FUZZY_ST            3             SO
KONSTANT            2             N
FUZZY_KO            3             N
**********************************************************************
**********************************************************************

 ****  RELATIONENKONTROLLBLOCK :TEILE_K   .RCB ****

**********************************************************************

ATTRIBUTNAME      FORMAT      SCHLUESSEL
 ---------------------------------------

NAME                3             N
EINZELTEIL          3             P
MODAL               1             N
VERBUNDEN           3             N
ZUSHANG_K           3             N
**********************************************************************
```

```
**********************************************************************

 ****  RELATIONENKONTROLLBLOCK :VERALLZYLK.RCB ****

**********************************************************************

ATTRIBUTNAME      FORMAT      SCHLUESSEL
 ---------------------------------------

NAME                3             P
EINHEIT             2             N
LAENGE_A            2             SO
FUZZY_A             3             N
GRUNDFLAEK          3             N
DECKFLAEK           3             N
SWEEPINGK           3             N
BED_1               4             N
PAR_11              3             N
PAR_12              3             N
BED_2               4             N
PAR_21              3             N
PAR_22              3             N
BED_3               4             N
PAR_31              3             N
PAR_32              3             N
BED_4               4             N
PAR_41              3             N
PAR_42              3             N
**********************************************************************

**********************************************************************

 ****  RELATIONENKONTROLLBLOCK :ZUSHANG_K .RCB ****

**********************************************************************

ATTRIBUTNAME      FORMAT      SCHLUESSEL
 ---------------------------------------

NAME                3             P
URSPRUNG_X          2             N
FUZZY_UX            3             N
URSPRUNG_Y          2             N
FUZZY_UY            3             N
URSPRUNG_Z          2             N
FUZZY_UZ            3             N
ROTATION_X          2             N
FUZZY_RX            3             N
ROTATION_Y          2             N
FUZZY_RY            3             N
ROTATION_Z          2             N
FUZZY_RZ            3             N
**********************************************************************
```

```
**************************************************************************

 ****  RELATIONENKONTROLLBLOCK :REGIO_K    .RCB ****

**************************************************************************

ATTRIBUTNAME      FORMAT      SCHLUESSEL
 ---------------------------------------

NAME                3             P
INTENSIT            1             N
FUZZY_IN            3             N
FARBE               3             N
TEXTUR              3             N
**************************************************************************
```

**** RELATIONENKONTROLLBLOCK :FLAECHEN .RCB ****

ATTRIBUTNAME	FORMAT	SCHLUESSEL
FNAME	3	P
NAME1	3	P
NAME2	3	N
NAME3	3	N
NAME4	3	N
NAME5	3	N
NAME6	3	N
NAME7	3	N
NAME8	3	N
NAME9	3	N
NAMEA	3	N
NAMEB	3	N
NAMEC	3	N
NAMED	3	N
NAMEE	3	N

**** RELATIONENKONTROLLBLOCK :GRUNDF .RCB ****

ATTRIBUTNAME	FORMAT	SCHLUESSEL
NAME	3	P
FORM	3	N
LOK_DURCH	2	N
LOK_X1	2	N
LOK_Y1	2	N
LOK_X2	2	N
LOK_Y2	2	N
LOK_X3	2	N
LOK_Y3	2	N
LOK_X4	2	N
LOK_Y4	2	N

```
**************************************************************************

 ****  RELATIONENKONTROLLBLOCK :OBERFARB  .RCB ****

**************************************************************************
ATTRIBUTNAME      FORMAT      SCHLUESSEL
 ----------------------------------------

NAME                3             P
ROT                 1             N
GRUEN               1             N
BLAU                1             N
**************************************************************************

**************************************************************************

 ****  RELATIONENKONTROLLBLOCK :REGIO     .RCB ****

**************************************************************************

ATTRIBUTNAME      FORMAT      SCHLUESSEL
 ----------------------------------------

NAME                3             P
GRUNDF              3             N
OBERFARB            3             N
TEXTUR              3             N
INTENSIT            1             SO
ZUSH                3             N
**************************************************************************

**************************************************************************

 ****  RELATIONENKONTROLLBLOCK :SWEEP     .RCB ****

**************************************************************************

ATTRIBUTNAME      FORMAT      SCHLUESSEL
 ----------------------------------------

NAME                3             P
CHARAKT             3             N
STEIGUNG            2             N
KONSTANT            2             N
**************************************************************************
```

```
**********************************************************************

 ****  RELATIONENKONTROLLBLOCK :VERZYL    .RCB ****

**********************************************************************

ATTRIBUTNAME      FORMAT      SCHLUESSEL
 ---------------------------------------

NAME                3             P
EINHEIT             2             N
LAENGE_A            2             SO
GRUNDF              3             N
DECKF               3             N
SWEEP               3             N
ZUSH                3             N
**********************************************************************

**********************************************************************

 ****  RELATIONENKONTROLLBLOCK :VERZYLAT  .RCB ****

**********************************************************************

ATTRIBUTNAME      FORMAT      SCHLUESSEL
 ---------------------------------------

NAME                3             P
OBERFARB            3             N
INTENSIT            1             SO
TEXTUR              3             N
OBERNUMM            1             P
**********************************************************************

**********************************************************************

 ****  RELATIONENKONTROLLBLOCK :ZUSH      .RCB ****

**********************************************************************

ATTRIBUTNAME      FORMAT      SCHLUESSEL
 ---------------------------------------

NAME                3             P
URSPRUNG_X          2             N
URSPRUNG_Y          2             N
URSPRUNG_Z          2             N
ROTATION_X          2             N
ROTATION_Y          2             N
ROTATION_Z          2             N
**********************************************************************
```

Anhang C1: Lokale Bewertungen mit erhöhtem Minimum

Zyl.(Modell)	Zyl.(Bild)	Vergl.Wert
VZ1	VVZ1	1.000000
VZ1	VVZ2	0.562500
VZ1	VVZ3	0.562500
VZ1	VVZ4	0.500000
VZ1	VVZ5	0.031250
VZ1	VVZ7	0.000000
VZ1	VVZ8	0.000000
VZ2	VVZ1	0.500000
VZ2	VVZ2	0.031250
VZ2	VVZ3	0.031250
VZ2	VVZ4	1.000000
VZ2	VVZ5	0.845703
VZ2	VVZ7	0.000000
VZ2	VVZ8	0.000000
VZ3	VVZ1	0.562500
VZ3	VVZ2	1.000000
VZ3	VVZ3	1.000000
VZ3	VVZ4	0.031250
VZ3	VVZ5	0.062500
VZ3	VVZ7	0.000000
VZ3	VVZ8	0.000000
VZ4	VVZ1	0.031250
VZ4	VVZ2	0.062500
VZ4	VVZ3	0.062500
VZ4	VVZ4	0.023926
VZ4	VVZ5	0.500000
VZ4	VVZ7	0.000000
VZ4	VVZ8	0.000000
VZ5	VVZ1	0.562500
VZ5	VVZ2	1.000000
VZ5	VVZ3	1.000000
VZ5	VVZ4	0.031250
VZ5	VVZ5	0.062500
VZ5	VVZ7	0.000000
VZ5	VVZ8	0.000000
VZ6	VVZ1	0.562500
VZ6	VVZ2	1.000000
VZ6	VVZ3	1.000000
VZ6	VVZ4	0.031250
VZ6	VVZ5	0.062500
VZ6	VVZ7	0.000000
VZ6	VVZ8	0.000000
VZ7	VVZ1	0.031250
VZ7	VVZ2	0.062500
VZ7	VVZ3	0.062500
VZ7	VVZ4	0.023926
VZ7	VVZ5	0.500000
VZ7	VVZ7	0.000000
VZ7	VVZ8	0.000000

VZ8	VVZ1	1.000000
VZ8	VVZ2	0.562500
VZ8	VVZ3	0.562500
VZ8	VVZ4	0.031250
VZ8	VVZ5	0.031250
VZ8	VVZ7	0.000000
VZ8	VVZ8	0.000000
VZ9	VVZ1	0.250000
VZ9	VVZ2	0.031250
VZ9	VVZ3	0.031250
VZ9	VVZ4	0.031250
VZ9	VVZ5	0.023926
VZ9	VVZ7	0.000000
VZ9	VVZ8	0.000000
VZ10	VVZ1	0.031250
VZ10	VVZ2	0.062500
VZ10	VVZ3	0.062500
VZ10	VVZ4	0.845703
VZ10	VVZ5	1.000000
VZ10	VVZ7	0.000000
VZ10	VVZ8	0.000000
VZ11	VVZ1	0.562500
VZ11	VVZ2	0.562500
VZ11	VVZ3	0.562500
VZ11	VVZ4	0.250000
VZ11	VVZ5	0.187500
VZ11	VVZ7	0.000000
VZ11	VVZ8	0.000000
VZ12	VVZ1	0.002197
VZ12	VVZ2	0.002197
VZ12	VVZ3	0.383789
VZ12	VVZ4	0.002197
VZ12	VVZ5	0.000732
VZ12	VVZ7	0.000000
VZ12	VVZ8	0.000000
VZ13	VVZ1	0.562500
VZ13	VVZ2	1.000000
VZ13	VVZ3	1.000000
VZ13	VVZ4	0.031250
VZ13	VVZ5	0.062500
VZ13	VVZ7	0.000000
VZ13	VVZ8	0.000000
VZ14	VVZ1	0.562500
VZ14	VVZ2	1.000000
VZ14	VVZ3	1.000000
VZ14	VVZ4	0.031250
VZ14	VVZ5	0.062500
VZ14	VVZ7	0.000000
VZ14	VVZ8	0.000000

VZ15	VVZ1	0.562500
VZ15	VVZ2	1.000000
VZ15	VVZ3	1.000000
VZ15	VVZ4	0.031250
VZ15	VVZ5	0.062500
VZ15	VVZ7	0.000000
VZ15	VVZ8	0.000000
VZ16	VVZ1	0.562500
VZ16	VVZ2	1.000000
VZ16	VVZ3	1.000000
VZ16	VVZ4	0.031250
VZ16	VVZ5	0.062500
VZ16	VVZ7	0.000000
VZ16	VVZ8	0.000000
VZ17	VVZ1	0.000000
VZ17	VVZ2	0.000000
VZ17	VVZ3	0.000000
VZ17	VVZ4	0.000000
VZ17	VVZ5	0.000000
VZ17	VVZ7	1.000000
VZ17	VVZ8	0.833333
VZ18	VVZ1	0.000000
VZ18	VVZ2	0.000000
VZ18	VVZ3	0.000000
VZ18	VVZ4	0.000000
VZ18	VVZ5	0.000000
VZ18	VVZ7	0.250000
VZ18	VVZ8	0.500000

Anhang C2: Lokale Bewertungen mit Mittelwert

Zyl.(Modell)	Zyl.(Bild)	Vergl.Wert
VZ1	VVZ1	1.000000
VZ1	VVZ2	0.529412
VZ1	VVZ3	0.470588
VZ1	VVZ4	0.705882
VZ1	VVZ5	0.352941
VZ1	VVZ7	0.000000
VZ1	VVZ8	0.000000
VZ2	VVZ1	0.805882
VZ2	VVZ2	0.411765
VZ2	VVZ3	0.411765
VZ2	VVZ4	1.000000
VZ2	VVZ5	0.435294
VZ2	VVZ7	0.000000
VZ2	VVZ8	0.000000
VZ3	VVZ1	0.470588
VZ3	VVZ2	0.705882
VZ3	VVZ3	0.764706
VZ3	VVZ4	0.352941
VZ3	VVZ5	0.529412
VZ3	VVZ7	0.000000
VZ3	VVZ8	0.000000
VZ4	VVZ1	0.352941
VZ4	VVZ2	0.529412
VZ4	VVZ3	0.470588
VZ4	VVZ4	0.352941
VZ4	VVZ5	0.705882
VZ4	VVZ7	0.000000
VZ4	VVZ8	0.000000
VZ5	VVZ1	0.470588
VZ5	VVZ2	0.941176
VZ5	VVZ3	0.764706
VZ5	VVZ4	0.352941
VZ5	VVZ5	0.529412
VZ5	VVZ7	0.000000
VZ5	VVZ8	0.000000
VZ6	VVZ1	0.470588
VZ6	VVZ2	0.764706
VZ6	VVZ3	0.941176
VZ6	VVZ4	0.411765
VZ6	VVZ5	0.470588
VZ6	VVZ7	0.000000
VZ6	VVZ8	0.000000
VZ7	VVZ1	0.352941
VZ7	VVZ2	0.470588
VZ7	VVZ3	0.529412
VZ7	VVZ4	0.294118
VZ7	VVZ5	0.764706
VZ7	VVZ7	0.000000
VZ7	VVZ8	0.000000

VZ8	VVZ1	0.705882
VZ8	VVZ2	0.470588
VZ8	VVZ3	0.470588
VZ8	VVZ4	0.470588
VZ8	VVZ5	0.352941
VZ8	VVZ7	0.111111
VZ8	VVZ8	0.111111
VZ9	VVZ1	0.632353
VZ9	VVZ2	0.367647
VZ9	VVZ3	0.352941
VZ9	VVZ4	0.470588
VZ9	VVZ5	0.294118
VZ9	VVZ7	0.111111
VZ9	VVZ8	0.111111
VZ10	VVZ1	0.411765
VZ10	VVZ2	0.529412
VZ10	VVZ3	0.470588
VZ10	VVZ4	0.411765
VZ10	VVZ5	0.941176
VZ10	VVZ7	0.000000
VZ10	VVZ8	0.000000
VZ11	VVZ1	0.470588
VZ11	VVZ2	0.470588
VZ11	VVZ3	0.470588
VZ11	VVZ4	0.470588
VZ11	VVZ5	0.470588
VZ11	VVZ7	0.000000
VZ11	VVZ8	0.000000
VZ12	VVZ1	0.294118
VZ12	VVZ2	0.294118
VZ12	VVZ3	0.352941
VZ12	VVZ4	0.235294
VZ12	VVZ5	0.235294
VZ12	VVZ7	0.111111
VZ12	VVZ8	0.111111
VZ13	VVZ1	0.529412
VZ13	VVZ2	0.941176
VZ13	VVZ3	0.941176
VZ13	VVZ4	0.352941
VZ13	VVZ5	0.470588
VZ13	VVZ7	0.074074
VZ13	VVZ8	0.111111
VZ14	VVZ1	0.529412
VZ14	VVZ2	1.000000
VZ14	VVZ3	0.941176
VZ14	VVZ4	0.411765
VZ14	VVZ5	0.470588
VZ14	VVZ7	0.000000
VZ14	VVZ8	0.000000
VZ15	VVZ1	0.529412
VZ15	VVZ2	0.764706
VZ15	VVZ3	0.941176
VZ15	VVZ4	0.411765
VZ15	VVZ5	0.470588
VZ15	VVZ7	0.000000
VZ15	VVZ8	0.000000

VZ16	VVZ1	0.470588
VZ16	VVZ2	0.705882
VZ16	VVZ3	1.000000
VZ16	VVZ4	0.352941
VZ16	VVZ5	0.529412
VZ16	VVZ7	0.000000
VZ16	VVZ8	0.000000
VZ17	VVZ1	0.000000
VZ17	VVZ2	0.000000
VZ17	VVZ3	0.000000
VZ17	VVZ4	0.000000
VZ17	VVZ5	0.000000
VZ17	VVZ7	1.000000
VZ17	VVZ8	0.555556
VZ18	VVZ1	0.000000
VZ18	VVZ2	0.000000
VZ18	VVZ3	0.000000
VZ18	VVZ4	0.000000
VZ18	VVZ5	0.000000
VZ18	VVZ7	0.333333
VZ18	VVZ8	0.666667

Lebenslauf

Geburtsdatum:	26. Oktober 1956 in Nürnberg
Familienstand:	verheiratet
1963 - 1967	Volksschule Nürnberg
1967 - 1976	Neues Gymnasium Nürnberg, Abschluß Allgemeine Hochschulreife
1976 - 1981	Studium der Informatik an der Friedrich-Alexander Universität Erlangen-Nürnberg
Juni 1980	Summer School on Digital Signal Processing bei CISM, Udine, als Stipendiat
1981 - 1982	wissenschaftlicher Angestellter am Lehrstuhl für Technische Elektronik (Prof. Dr. Seitzer) der FAU Erlangen-Nürnberg
1982 - 1986	wissenschaftlicher Angestellter am Lehrstuhl für Informatik 5 (Mustererkennung) (Prof. Dr. Niemann) der FAU Erlangen-Nürnberg
seit September 1986	Entwicklungsingenieur bei der Fa. Siemens in Erlangen

In letzter Zeit sind folgende Arbeitsberichte des Instituts für Mathematische Maschinen und Datenverarbeitung erschienen:

Band 14

Nr. 1 Pattern Recognition
(Research at Lehrstuhl für Informatik 5)
(Februar 1981)

Nr. 2 Bunke, H.:
Analyse elektrischer Schaltpläne
(März 1981)

Nr. 3 Linster, C.-U.:
SYMPOS/UNIX - Ein Betriebssystem für homogene Polyprozessorsysteme
(Juni 1981)

Nr. 4 Akyildiz, J.F., Bolch, G.:
Analytic Solution Techniques for Queueing Network Models of Computer Systems
(Juli 1981)

Nr. 5 Lindstedt, W.G.:
Ein Verfahren zur Erstellung portabler algorithmischer Laufzeitprogramme
(Juli 1981)

Nr. 6 Vollmar, R.:
Zum Begriff des Parallelismus bei Polyautomaten
(Juli 1981)

Nr. 7 Luft, A.L.:
Rationaler Sprachgebrauch und orthosprachliche Standardisierung als Grundlagen zuverlässiger Software-Entwicklung und -Dokumentation
(Januar 1982)

Nr. 8 Händler, W.:
PROCEEDINGS OF A WORKSHOP ON TAXONOMY IN COMPUTER ARCHITECTURE
(Februar 1982)

Nr. 9 Beth, Th., Hess, P., Wirl, K.:
Materialien zur Kryptographie
(März 1982)

Nr. 10 Jahresbericht 1981 der Informatik
(März 1982)

Band 15
=======

Nr. 1 Wittmann, A.:
Ein Mechanismus für die Synchronisation paralleler Prozesse
(Februar 1982)

Nr. 2 Maehle, E.:
Fehlertolerantes Verhalten in Multiprozessoren - Untersuchungen zur Diagnose und Rekonfiguration
(März 1982)

Nr. 3 Bathelt, P.:
Vergleich von Synchronisationsmechanismen
(Mai 1982)

Nr. 4 Keramidis, S.:
Eine Methode zur Spezifikation und korrekten Implementierung von asynchronen Systemen
(Juni 1982)

Nr. 5 Fromm, H.J.:
Multiprozessor-Rechenanlagen:
Programmstrukturen, Maschinenstrukturen und Zuordnungsprobleme
(Mai 1982)

Nr. 6 Bley, H.:
Vorverarbeitung und Segmentierung von Stromlaufplänen unter Verwendung von Bildgraphen
(Juli 1982)

Nr. 7 Hchl, W.:
Informatik-Sammlung
Katalog mit historischen Erläuterungen
(August 1982)

Nr. 8 Messerer, M.:
Ein neuer Ansatz zur Parallelisierung von Compilern
(August 1982)

Nr. 9 Grosch, J.:
Eine Programmiersprache mit mengentheoretischen Konstrukten und deren effiziente Implementierung
(August 1982)

Nr. 10 Kleinöder, W.:
Stochastische Bewertung von Aufgabenstrukturen für hierarchische Mehrrechnersysteme
(August 1982)

Band 15 Fortsetzung
==========

Nr. 11 Kneißl, F.:
Realisierung von Makro-Datenflußmechanismen auf hierarchischen Mehrrechnersystemen
(August 1982)

Nr. 12 Brendel, W., Fan, Z., Klar, R., Schmielau, W.:
ERES 82
Handbuch und Fallstudie
Handbook and Case Study
(Oktober 1982)

Nr. 13 Pelz, K.:
Portable Codegenerierung in Übersetzern für Prozeß-Programmiersprachen
(Oktober 1982)

Nr. 14 Sommer, U., Steinchen, J.:
Ein Kleinrechner-Grafiksystem zur Dokumentation von EDV-Abläufen
(Oktober 1982)

Nr. 15 Hein, H.-W.:
Das Erlanger Spracherkennungssystem
(November 1982)

Nr. 16 Bolch, G., Bruchner, W.:
Analytische Modelle für symmetrische Mehrprozessoranlagen mit dynamischen Prioritäten
(Dezember 1982)

Nr. 17 Jahresbericht 1982 der Informatik
(März 1983)

Band 16
=======

Nr. 1 Gall, R.:
Formale Beschreibung des inkrementellen Programmierens-im-Großen mit Graph-Grammatiken
(März 1983)

Nr. 2 König, R.:
Beiträge zur algebraischen Theorie der formalen Sprachen
(März 1983)

Nr. 3 Welbourn, D.B.:
The Design of Mechanical Components and the Development of DUCT:
17 Years of CAD/CAM in Cambridge University
(April 1983)

Nr. 4 Woitok, R.:
Die System-Implementierungs-Sprache ALICE
(Mai 1983)

Nr. 5 Schneider, S., Schwab, P., Renninger, W.:
Wesen, Vergleich und Stand von Software zur Produktion von Systemen der computergestützten Unternehmensplanung
(Juli 1983)

Nr. 6 Händler, W., Herzog, U., Hofmann, F., Schneider, H.J. und Mitarbeiter:
Projektabschlußbericht 1978 - 1982 Erlangen General Purpose Array
(Juli 1983)

Nr. 7 Mackert, L.:
Modellierung, Spezifikation und korrekte Realisierung von asynchronen Systemen
(Juli 1983)

Nr. 8 Reitenspieß, M.:
Sprachkonstrukte zur Spezifikation und korrekten Implementierung von Schutzproblemen
(Juli 1983)

Nr. 9 Weber, K.:
Modellierung von Fehlverhalten mit Berücksichtigung paralleler Abläufe
(August 1983)

Nr. 10 Dietsch, H.:
Ein verteiltes Kommunikationssystem für einen Verbund aus lokalen Mikrorechner-Netzwerken
(Dezember 1983)

Band 16 Fortsetzung
==========

Nr. 11 Maehle, E., Schmitter, E.:
Workshop Fehlertolerante Mehrprozessor- und Mehrrechnersysteme
(Dezember 1983)

Nr. 12 Volkert, J.:
Multiplikative Abstandsfunktionen und deren Anwendung in der Bildverarbeitung
(Dezember 1983)

Nr. 13 Steinbauer, D.:
Transaktionen als Grundlage zur Strukturierung und Integritätssicherung in Datenbank-Anwendungssystemen
(Dezember 1983)

Nr. 14 Wilke, P.:
Zusammenhänge und Unterschiede zwischen Graph-Grammatiken und Petri-Netzen sowie verwandter Systeme
(Dezember 1983)

Nr. 15 Jahresbericht der Informatik 1983
(März 1984)

Band 17
=======

Nr. 1 Eberlein, W.:

Architektur technischer Datenbanken für integrierte Ingenieursysteme
(Februar 1984)

Nr. 2 Mertens, P., Heigl, M.:

Neuere Entwicklungen der computergestützten Produktionsplanung, Eignung - Verbindungen - Entwicklungspfade
(April 1984)

Nr. 3 Stoyan, H.:

Maschinen- unabhängige Code-Erzeugung als semantikerhaltende beweisbare Programmtransformation
(Juli 1984)

Nr. 4 Brendel, W.:

Funktionaler Entwurf und automatische Synthese hochintegrierter Schaltkreise
(August 1984)

Nr. 5 Brietzmann, A.:

Semantische und pragmatische Analyse im Erlanger Spracherkennungsprojekt
(Oktober 1984)

Nr. 6 Hoppe, B.:

Deterministisches Zuordnen vielfach durchlaufener Aufgabensysteme in Multiprozessorsystemen
(Oktober 1984)

Nr. 7 Fumy, W., Posch, S., Rieß, H.P.:

Materialien zur Codierungstheorie III
(Dezember 1984)

Nr. 8 Akyildiz, J. F.:

Leistungsanalyse von Multiprozessorsystemen mit Prozeßkommunikation
(Dezember 1984)

Nr. 9 Jahresbericht der Informatik 1984

(März 1985)

Band 18
=======

Nr. 1 Siegmann, H.:

Wesen, Vergleich und Stand von zehn ausgewählten kleinrechnerorientierten Planungssprachen
(Februar 1985)

Nr. 2 Rathke, M.:

Parallelisieren ordnungserhaltender Programmsysteme für hierarchische Multiprozessorsysteme
(Juni 1985)

Nr. 3 Geus, L.:

Parallelisierung eines Mehrgitterverfahrens für die Navier-Stokes-Gleichungen auf EGPA-Systemen
(Juli 1985)

Nr. 4 Hellmold, K.-U.:

Der Simulationsrechner SIMPLEX
Eine Multiprozessorkonfiguration zur Simulation von zeitdiskreten Systemen auf GPSS-FORTRAN-Basis
(September 1985)

Nr. 5 Klebes, G.:

Entwicklung eines botschaftsorientierten Synchronisationsverfahrens für parallele Prozesse und seine Realisierung im Rahmen eines Realzeitprogrammiersystems
und
Rupprecht, G.:

Ein Konzept zur Integration von allgemeinen Sprachkonstrukten zur Synchronisation und von gemeinsamen abstrakten Datentypen mehrerer Prozesse in Modula-2 und seine Implementierung

Nr. 6 Schön, D.:

Technisches Informationssystem und Entwurf integrierter Ingenieurdatenbanken
(Oktober 1985)

Nr. 7 Vasconcelos, A.J.V. de:

Berechnung nichtstationärer Vorgänge in Drehstromnetzen mit Raumzeigerkomponenten auf einem Multiprozessor
(Dezember 1985)

Nr. 8 Baader, F.:

Die S-Varietät DS und einige Untervarietäten
(Dezember 1985)

Nr. 9 Jahresbericht der Informatik 1985
(März 1986)

Band 19
=======

Nr. 1 Erhard, W.:

Feldrechner DAP: Invertierung großer Matrizen
(Februar 1986)

Nr. 2 Fumy, W.:

Über orthogonale Transformationen und fehlerkorrigierende Codes
(März 1986)

Nr. 3 Ehrlich, U.:

Ein Lexikon für das natürlich-sprachliche Dialogsystem EVAR
(Mai 1986)

Nr. 4 Mühlfeld, R.:

Verifikation von Worthypothesen
(Juni 1986)

Nr. 5 Blank, E., Bolch, G.:

Leistungsbewertung fehlertoleranter Systeme
(Juni 1986)

Nr. 6 Mertens,P., Allgeyer, K., Däs, H., Schumann, M.:

Betriebliche Expertensysteme in deutschsprachigen Ländern
- Versuch einer Bestandsaufnahme -
(Juli 1986)

Nr. 7 Moritzen, K.:

Softwarehilfsmittel zur Programmierung von Multiprozessoren mit begrenzten Nachbarschaften

Ein Beitrag zur Konfigurationsverwaltung
(September 1986)

Nr. 8 Kragl, G.:

Eine Programmierumgebung für verteiltes PEARL-Codeerzeugung mittels Programmsynthese

(Oktober 1986)

Nr. 9 Völk, A.:

Ein vollständiges Mengenkonzept in Programmiersprachen - Realisierung mit unkonventioneller Hardware

(Oktober 1986)

Band 19 Fortsetzung
==========

Nr. 10 Kratzer, K.:
Komponenten der Datenverwaltung in der Büroorganisation
(Dezember 1986)

Nr. 11 Bathelt, P.:
Der Dominoeffekt in der Fehlerbehandlung von
Prozeß-Systemen
(Dezember 1986)

Nr. 12 Karl, W.:
Erstellung eines Konzeptes zur Mikroprogrammierung
eines Koprozessors für schnelle Arithmetik
(Dezember 1986)

Nr. 13 Jahresbericht 1986 der Informatik
(März 1987)

Offsetdruck · Fotodruck · Hogl · Hauptstraße 109 · 8520 Erlangen · Tel.09131/22990